UNIVERSIDAD DEL VALLE-CALI-COLOMBIA
Escuela de Ingeniería de Sistemas y Computación

Tésis de Maestría

Para obtener el título de

Magíster en Ingeniería de Sistemas y Computación

Autor

ALVARO RICARDO CUJAR ROSERO

POS TAGGING UTILIZANDO CRFs PARA EL ESPAÑOL

Director

Raul Ernesto Gutierrez de Piñeres Reyes Ph.D.

2016

Al Padre eterno,
y a la persona que ha entregado su vida por mí: mi hermosa mamá Ligia,
y a mi futura esposa Margarita.

Resumen

El POS Tagging es una tarea enmarcada dentro del procesamiento del lenguaje Natulal (PLN), cuyo objetivo esencial es la categorización lingüística de las palabras dentro de un texto. El problema al cual se enfrenta el POS tagging es la ambigüedad, en cuanto a que existen algunas palabras que presentan más de una posible categoría lingüística. Existen diferentes enfoques para resolver esta tarea, entre los que se tienen, modelos estocásticos, modelos basados en reglas, entre otros. Dentro de los modelos estocásticos se encuentran los campos randómicos condicionales (Conditional Random Fields CRF) y las cadenas ocultas de markov (Hidden Markov Chain HMM). Los primeros, utilizados para el etiquetado de secuencias, representan un modelo gráfico no dirigido, lo que permite que sean discriminativos, y que se puedan incluir características sobre las observaciones. Los segundos, igualmente utilizados para el etiquetado de secuencias, corresponden a un modelo gráfico dirigido, lo cual permite que sean generativos. La desventaja que presentan las HMMs, es que elas para determinar la clasificación del siguiente estado, no tienen en cuenta el pasado, sino el estado actual, además de no permitir incluir características sobre las observaciones. Estas dos debilidades no las presenta los CRFs. El objetivo de este trabajo es realizar un análisis comparativo del rendimiento de estos dos modelos estocásticos, para la tarea del POS Tagging, en el idioma español, bajo el corpus AnCora, teniendo en cuenta las características morfológicas de cada palabra para el modelo CRF.

Agradecimientos

Agradecimiento total al Padre Eterno. Él siempre, desde el primer día, ha estado a mi lado en este proceso de formación de Magister y ha permitido que yo supere cada uno de los obstáculos que se me han presentado.

Este trabajo no habría sido posible sin la infaltable e incondicional colaboración del Doctor Raúl Ernesto Gutierrez de Piñeres, quien siempre me orientó, supervisó y aconsejó. Él en todo momento, estuvo disponible, a pesar de sus numerosos compromisos, para resolver dudas y guiarme, depositando en mí toda su gran experiencia, conocimiento y confianza.

Agradezco a mi Familia. Su incondicional apoyo en todo momento. Sus fuerzas, sus ánimos, su entrega total para conmigo en este proyecto, han hecho que me alimente de fortaleza cada día.

De igual forma agradezco a mi prometida Margarita Portilla, quien en todos los instantes me dió su fortaleza, su oración, su energía positiva y su amor para poder culminar este trabajo.

Agradecimientos a la Universidad Mariana, la cual me apoyó con asignación de carga académica para poder finalizar este trabajo.

Finalmente mis agradecimientos al profesor Oscar Orlando Ceballos y al candidato a Magister, Christian Victoria, quienes, a pesar de sus ocupaciones, me dedicaron parte de su tiempo en algunas explicaciones.

Índice general

1. Introducción

El Procesamiento de Lenguaje Natural o PLN, es un área de conocimiento que ha tomado fuerza en las últimas décadas, que tiene como fin procesar de texto, mediante la realización de diversas tareas, tales como el análisis morfológico, etiquetado de categorías (en adelante POS Tagging, por su sigla en inglés de Part of Speech Tagging), análisis sintáctico, semántico, dicursivo, etc. Estas tareas se realizan de una forma automática, usando técnicas de machine learning.

En este contexto, una de las tareas del PLN es el POS Tagging, el cual se encarga de asignar a cada palabra de un texto dado, la etiqueta correspondiente a la categoría lingüística adecuada. Estas categorías son: adjetivos, adverbios, determinantes, preposiciones, pronombres, sustantivos y verbos entre otras. En la oración 'El carro es bonito', la palabra 'el' es etiquetada como determinante, la palabra 'carro' es etiquetada como sustantivo, la palabra 'es' es etiquetado como verbo y la palabra 'bonito' es etiquetada como adjetivo.

La tarea del POS Tagging se puede formalizar como un problema de etiquetado secuencial, en el marco del aprendizaje supervisado, en la cual una sentencia de entrada, conformada por palabras, es anotada con etiquetas de categorías mediante una función de mapeo, además la tarea de POS tagging es una de las más importantes como soporte a otras tareas de PLN como el análisis sintáctico probabilístico, NER, NP-chunking, entre otros.

Las soluciones de software que permiten clasificar la categoría adecuada a cada palabra se denominan etiquetadores o POS Taggers. En este contexto, se pueden definir 3 enfoques con relación a la tarea de POS Tagging: modelos estocásticos, modelo basado en reglas y modelo basado en transformaciones. En este trabajo se tratará los modelos estocásticos.

Entre los modelos estocásticos se encuentran las Cadenas Ocultas de Markov (en adelante HMM, por su sigla en inglés de Hidden Markov Model), Modelos de Markov de Máxima Entropía (en adelante MEMM, por su sigla en inglés de Maximum Entropy Markov Model) y los Campos Randómicos Condicionales (en adelante CRFs, por su sigla en inglés de Conditional Random Fields) estos dos últimos permiten el etiquetado secuencial de manera discriminativa de las palabras que conforman

una sentencia a diferencia de los HMMs, que lo realizan de una forma generativa y que calculan la probabilidad condicional de manera indirecta. Entre otras, los CRFs permiten encontrar una distribución de probabilidad sobre un conjunto de variables representadas bajo un modelo gráfico. Se han aplicado los CRFs en la resolución de problemas de POS Tagging, NER, entre otros, teniendo mejores resultados comparados con los resultados obtenidos mediante la aplicación de métodos como HMMs y modelos basados en máxima entropía para el idioma inglés [24]. También en este trabajo se presenta un estudio comparativo del rendimiento de los modelos CRF y HMM, en la tarea de POS Tagging, utilizando el corpus AnCora para el idioma español, como contribución al NLP.

En la primera parte del documento se hace una introducción del trabajo, con el fin de explicar el proceso de desarrollo en forma general. En la segunda parte se encuentra la revisión de literatura que se tuvo en cuenta y que soportó la investigación, en la tercera parte se propone la arquitectura del clasificador, y en la última sección se presenta el diseño de los experiemtos junto con los resultados obtenidos por los clasificadores y su correspondiente análisis.

1.1. Planteamiento del Problema

La tarea de POS Tagging se resume en la categorización léxico-semántica de cada palabra de un texto dado. Existen palabras cuya escritura es idéntica, pero su significado es distinto. La categoría que le corresponde a cada palabra, depende del contexto en el cual es usada y las diferentes asociaciones que tienen las palabras dentro del corpus. Tradicionalmente una de las maneras de resolver este problema, es especificando la solución como un problema de etiquetado secuencial.

El problema del etiquerado secuencial puede ser abordado usando modelos estocásticos como los HMMs, MEMMs, y CRFs.

Los HMMs han sido utilizadas con éxito tanto en el inglés como en el español, como es el caso de entre ellos el POS Tagger TnT [11]. Experimentos en el idioma inglés, trabajando con el corpus Penn Treebank [26] y en el alemán con el corpus NEGRA[1], arrojan resultados cercanos al 97 % de presición [11]. En [20] se realiza

[1]disponible en http://www.coli.uni-saarland.de/projects/sfb378/negra-corpus/

el experimento de TnT sobre el idioma español utilizando el corpus CLic-TALP, obteniendo resultados cercanos al 96.5 % de precisión. Dentro del idioma español, se encuentran taggers, que utilizan el modelo de los HMMs, entre ellos se encuentra FreeLing [2] obteniendo 97 % de presición [44], o TreeTagger, el cual utiliza el Modelo de Markov [41], el cual también ha sido aplicado al español.

Por otra parte, se han realizado trabajos utilizando CRFs, tales como: [24] [48] [7] [27] [9] [34], en tareas del NLP, como lo es el etiquetado secuencial, incluyendo el POS Tagging. Por esta razón se hace necesario un estudio comparativo entre los modelos HMMs y CRF, en la tarea del POS Tagging para el Español.

Las HMMs son modelos generativos, hacen estrictas suposiciones de independencia [24]. Dentro de las HMMs el cálculo de la probabilidad $p(y_t)$ depende de $p(y_{t-1})$, es decir $p(y_t|y_{t-1})$, en otras palabras se asume que cada estado depende únicamente del estado inmediatamente anterior, siendo independiente de los estados $y_{t-2}, ..., y_2, y_1$. Con las HMMs también se asume que cada observación x_t, depende únicamente del estado y_t. Con el cálculo de las anteriores probabilidades, se puede obtener la probabilidad conjunta $p(y, x)$.

Las HMMs no tienen en cuenta características sobre las observaciones. Por otra parte las MEMMs si las toman en cuenta, siendo un modelo discriminativo, el cálculo de $p(y_t)$ depende de $p(y_{t-1}, o_t)$ y de las características para la observación. La debilidad encontrada [24] en las MEMMs hace referencia al sesgo en el etiquetado, en el cual las transiciones que salen de un estado a otro, no compiten con todas las transiciones en el modelo, sino con algunas.

Estas debilidades no las posee el modelo CRF [24], siendo un modelo gráfico discriminativo no dirigido, supera los puntos débiles presentados por los HMMs y los MEMMs. En este sentido se ha querido investigar sobre el etiquetado secuencial para el idioma Español, utilizando CRFs, dado que éste tiene en cuenta las características de las observaciones, que para este trabajo corresponde a las características morfológicas de cada palabra, además de utilizar todo el conjunto de observaciones por completo.

[2] http://nlp.lsi.upc.edu/freeling/

1.1.1. Desafíos

- Selección de características y extracción de las mismas mediante algoritmos de extracción de características morfológicas sobre el corpus AnCora.
- Implementación y uso de los HMMs, con el objeto de poder obtener las medidas de desempeño de este modelo frente al Español, con el corpus AnCora.
- Implementación y uso del modelo CRF, con base en las características seleccionadas, obteniendo las medidas de desempeño de este modelo frente al Español, utilizando el corpus AnCora.
- Análisis de rendimiento de los clasificadores basados en la precisión, el accuraccy, el recall, y el score FB1, para el idioma español, bajo el corpus AnCora.

1.1.2. Pregunta de Investigación

El problema se puede resumir en: ¿Se pueden mejorar las medidas de desempeño de accuraccy, precisión, recall y FB1 para la tarea del POS Tagging usando características morfológicas de la sentencia de entrada?

1.2. Objetivos

En la presente sección se establece el alcance de la investigación, el cual está direccionado bajo uno objetivo principal, y para alcanzarlo se deben cumplir antes, un conjunto de objetivos específicos.

1.2.1. Objetivo general

Evaluar el desempeño de los CRFs para la tarea de POS tagging en el idioma español.

1.2.2. Objetivos específicos

1. Implementar la fase de preprocesamiento de la sentencia de entrada.

2. Adaptar los algoritmos de análisis morfológico y POS Tagging con HMMs desde freeling bajo AnCora.

3. Especificar las características morfológicas y de contexto que se necesitan para la construcción del modelo de CRFs.

4. Desarrollar los algoritmos de extracción de características para el modelo de CRFs.

5. Desarrollar e implementar los algoritmos de POS Tagging bajo el modelo de CRFs.

6. Implementar los algoritmos de medidas de desempeño para los modelos de HMMs y CRFs.

7. Comparar y analizar resultados obtenidos por los dos modelos, teniendo en cuenta las medidas de desempeño.

1.2.3. Hipótesis

Mediante el uso de características morfológicas y locales, el modelo CRF aumenta la presición en el etiquetado de secuencias, en el idioma español, haciendo uso del corpus AnCora y se mejora con respecto al uso de los HMMs.

1.3. Visión general del desarrollo de la investigación

Desde el punto de vista metodológico, este trabajo se enfoca en el análisis del rendimiento del modelo CRF, mediante el uso de características morfológicas en el idioma español. En este trabajo se toman mediciones de las variables accuracy, precisión, recall y FB1, tanto con el modelo CRF, como con el modelo HMM, bajo el corpus AnCora, permitiendo las comparaciones de estos dos modelos.

Como se observa en la figura 1.1, el corpus se preprocesa, es decir se adecúa de acuerdo al formato necesario para cada modelo HMM y CRF. Seguidamente se obtiene el conjunto de entrenamiento y el conjunto de testeo. En el caso del modelo

HMM, se realiza el entrenamiento, obteniendo los parámetros del modelo y posteriormente el correspondiente testeo. Para la extracción de características del CRF se obtienen todas las características de cada una de las palabras, de todas las sentencias de cada conjunto. Características como, el tiempo, el género, el número y la persona. Se tiene en cuenta características como, si la palabra comienza con mayúscula, si la palabra contiene algún símbolo, si la palabra contiene algún dígito, el tamaño de la palabra. Con estas caracteristicas, la palabra en sí y la correspondiente categoría lingüística, se construye el archivo preprocesado, el cual contiene la plantillas que serán necesarias para la creación de las funciones características por parte del modelo CRF.

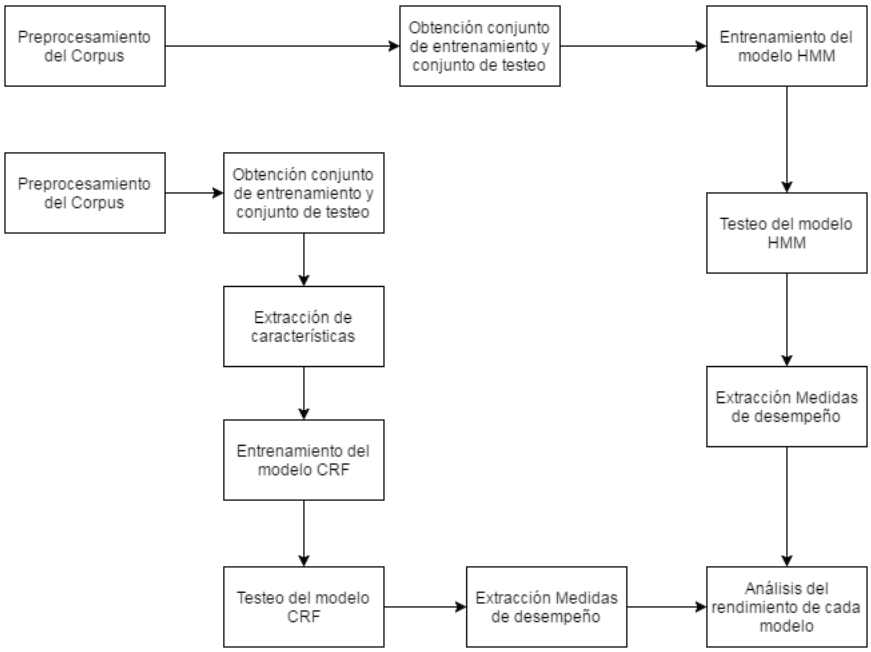


Figura 1.1: Desarrollo del trabajo

Los archivos preprocesados se entrenan y se obtiene un archivo que contiene todas las funciones características y los parámetros del modelo CRF, archivo mediante el cual ya se pueden realizar pruebas de testeo.

Una vez realizadas las pruebas de testeo para cada modelo, se calculan las medidas de desempeño, utilizando para ello las etiquetas asignadas por el modelo en el proceso

de etiquetado secuencial y las etiquetas que deberían ser correctas, las cuales se encuentran en el grupo de control, que corresponde al corpus AnCora. Finalmente, se realizan las comparaciones y el análisis de los resultados obtenidos.

1.4. Motivación

El POS Tagging es una de las técnicas del PLN más importantes y un paso obligatorio en todas las tareas relacionadas con el procesamiento de texto tales como el análisis sintáctico, NER, NP chunking entre otras.

En este documento, el POS Tagging es modelado como un problema de etiquetado secuencial e implementado con los CRFs, tomando como referencia el POS Tagging que realizan las HMMs y la presición que este modelo tiene sobre el español. El POS Tagging juega un papel muy importante, como por ejemplo, la no propagación de la ambigüedad de palabras a las posteriores tareas, en el procesamiento de cualquier texto.

De manera general, en [24] se presenta a los CRFs como modelos probabilísticos para la segmentación y el etiquetado de secuencias de datos, en donde, estos no tienen en cuenta suposiciones de independencia de las observaciones, dado que la probabilidad de una transición no depende únicamente de la observación actual, sino de las observaciones anteriores y posteriores, la cual es una limitación presente en los HMMs.

La motivación que permitió la realización de este trabajo radica en el uso de información morfológica y las características locales de la palabra, para resolver ambigüedad en la categorización de las palabras, además de poder realizar un análisis de desempeño de los modelos CRF y HMM, para el idioma Español, bajo el corpus AnCora, teniendo en cuenta que CRF utiliza, para el POS Tagging, todas las observaciones incluyendo las características que estas presentan, y HMM no las tiene en cuenta.

2. MARCO TEÓRICO

La tarea del POS Tagging es una de las tareas mas importantes del PLN y soporta otra serie de tareas como el análisis sintáctico probabilístico, análisis superficial sintáctico o chunking, reconocimiento de entidades, reconocimiento de relaciones entre entidades, entre otras. Los conceptos de etiquetado secuencial, el estudio de los modelos gráficos estocásticos, y la lingüística del corpus son el eje central teórico de este trabajo. En este capítulo se describe la tarea de POS Tagging como un problema de etiquetado secuencial y las posibles soluciones al mismo mediante el uso de modelos gráficos estocásticos.

2.1. Procesamiento de Lenguaje Natural PLN

El Procesamiento de Lenguaje Natural es un área de la Inteligencia Artificial que soporta las tareas de automatización del Lenguaje Natural usando modelos computacionales. Tareas tales como [23]:

- Agentes conversacionales o sistemas de diálogo: En donde se busca mantener diálogos con las personas. Esto implica que un sistema computacional soporte tanto, la entrada de lenguaje, como la salida del mismo. Dentro de la entrada de lenguaje está el reconocimiento automático del habla y del entendimiento del mismo. En la salida se encuentra generación de lenguaje natural y síntesis del mismo.
- Máquinas de traducción: Hace referencia a la tarea de poder recibir un texto en un idioma y tener como salida, el mismo texto en un idioma diferente.
- Web basada en "question answering": El uso del procesamiento de lenguaje natural para la realización de búsquedas web, en donde, en lugar de buscar por palabras aisladas, se puede buscar haciendo uso de preguntas estructuradas.

El procesamiento de lenguaje natural, no es una tarea trivial, de hecho, comprende un conjunto de tareas que orquestan entre ellas para lograr un determinado objetivo. Los procesos que se deben llevar a cabo en el PLN son los siguientes:

- Análisis morfológico: Por cada palabra se obtiene una salida con todas las características que esta suministra. Se divide la palabra en todos los morfemas que contenga ésta, luego con esa información se obtienen características como por ejemplo, la palabra es un sustantivo, un verbo o un adjetivo. También si la palabra está en plural o en singular, o si es un verbo conjugado y el tiempo del mismo.

- POS Tagging: Tarea relacionada con la categoraización de las palabras lingüísticamente. Se encarga de asignar a cada palabra de un texto dado, la etiqueta correspondiente a la catgoría lingüística. Estas categorías son: adjetivos, adverbios, determinantes, preposiciones, pronombres, sustantivos y verbos entre otras. Por ejemplo, a la palabra 'casa', se le otorga la etiqueta sustantivo.

- Análisis sintáctico: Se establece por cada oración la estructura sintáctica correspondiente, mediante la cual se determina si está bien escrita la oración o no, es decir si el orden de las palabras es el correcto.

- Análisis semántico: Proceso mediante el cual se obtiene el significado de una determinada sentencia, tomando como entrada el resultado suministrado por el análisis sintáctico.

- Análisis discursivo. Hace referencia al procesamiento de fragmentos de texto grandes. Dentro de este análisis se toma como un componente básico el análisis sintáctico y el análisis semántico, además de la pragmática la cual significa la intención del hablante.

En la actualidad el procesamiento de lenguaje natural no es un área exclusiva para la investigación; a nivel de la web son muchas las tareas que se soportan en el procesamiento de texto, minería de opinión, análisis de sentimientos, sistemas de pregunta respuesta, reconicimiento modal de textos en video, entre otros. La articulación de los algoritmos de búsqueda inteligente, los algoritmos de programación dinámica, la probabilidad estadística, la lingüística de corpus y el aprendizaje de máquina, han hecho del PLN un área de aplicación directa en muchas tareas como las mencionadas anteriormente. Quizás uno de los enfoques mas futuristas en pro del avance del PLN, es el uso de los CRFs como principal componente en la solución del

etiquetado secuencial. En tal sentido el PLN se orienta hacia las nuevas técnicas que presenta el aprendizaje de máquina.

2.2. Problema del POS Tagging

Como se mencionó en la sección anterior, el POS Tagging es una tarea en donde se categoriza cada palabra. El problema presente es la ambigüedad, que se encuentra cuando hay dos o mas palabras que se escriben idénticamente y su significado es distinto. En la oración 'El hombre bajo, tocó el bajo, bajo la escalera', como se puede observar en la figura 2.1, palabras como 'el', 'tocó' y 'escalera' que tienen una sola categoría lingüística. Palabras como 'hombre', 'bajo' y 'la' presentan más de una posible categoría lingüística. El POS tagging, se encarga de resolver este tipo de ambigüedades. Cuando dos o más palabras tienen igual escritura pero diferente etiqueta, se ejecutan mecanismos para resolver este problema, dentro de los cuales se encuentran los modelos estocásticos, sistemas basados en reglas y modelo basado en trasformaciones principalmente.

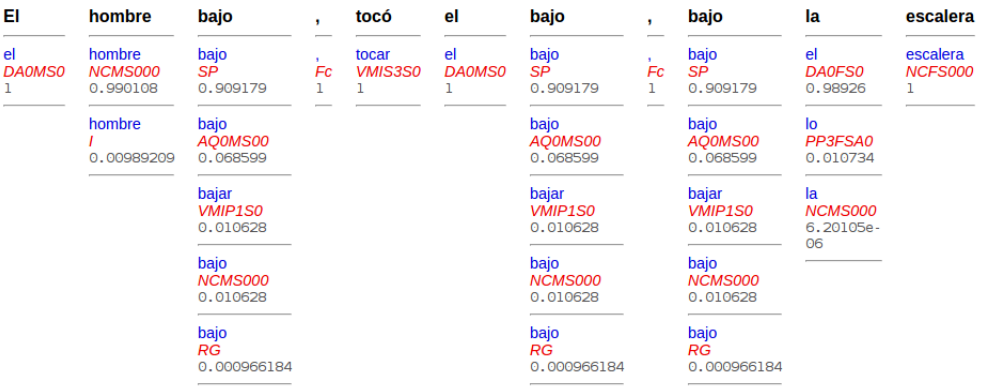


Figura 2.1: Ejemplo oración 'El hombre bajo, tocó el bajo, bajo la escalera', tomado desde freeling.

A continuación se presenta el resultado de ejecutar el POS tagging desde Freeling para la misma oración. Palabra 'El', determinante de tipo artículo, en masculino y

singular. Palabra 'hombre', sustantivo de tipo común, masculino y en singular. Palabra 'bajo', adjetivo calificativo cuyo género es masculino y se encuentra en singular. Palabra 'tocó', verbo de tipo principal, en modo indicativo, en pasado, en tercera persona y en singular. Palabra 'el', determinante de tipo artículo, en masculino y singular. Palabra 'bajo', sustantivo de tipo común, masculino y en singular. Palabra 'bajo', preposición simple. Palabra 'la', determinante de tipo artículo, en femenino y singular. Palabra 'escalera', sustantivo de tipo común, femenino y en singular.

2.3. Etiquetado de secuencias

Una secuencia corresponde a un conjunto de datos que guardan una relación entre sí [6]. Cada elemento de la secuencia tiene relación con sus vecinos directos, es decir, con los elementos que estan al lado del mismo.

El etiquetado de secuencias, hace referencia a un problema de clasificación, el cual se encuentra dentro del Machine learning, específicamente, aprendizaje supervisado, en donde se cuenta, inicialmente, con un conjunto de entrenamiento, y a partir del mismo se debe construir un clasificador, mediante el cual se puedan clasificar nuevas secuencias, es decir, a cada elemento observable de la secuencia se le otorga una determinada clase.

Problema del etiquetado secuencial.

Formalmente se plantea el problema del etiquetado secuencial de la siguiente forma: Sea L un alfabeto de etiquetas. Sea $Z = L^*$, es decir todas las posibles combinaciones de etiquetas. Sea X un conjunto de todas las posibles secuencias de entrada. Sea S un conjunto de entrenamiento compuesto por parejas $(\mathbf{x}, \mathbf{z})$, tomadas del espacio $D_{X \times Z}$. La secuencia $z = (z_1, z_2, z_3, ..., z_U)$ y la secuencia $x = (x_1, x_2, x_3, .., x_T)$, donde $|z| = U <= |x| = T$ y cada etiqueta z_i corresponde con un elemento x_i. El objetivo del etiquetado secuencial es utilizar al conjunto S para entrenar un algoritmo $h : X- > Z$, que permita etiquetar secuencias en un conjunto $S' \subset D_{X \times Z}$ disyunto de S [4].

Visto una manera mas simple, [19] define el etiquetado secuencial de la siguiente forma: se cuenta con un conjunto de $\{(x_i, y_i)\}_{i=1}^{N}$, de N ejemplos, en donde $x_i = \langle \mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{i,T_i} \rangle$ e $yi = \langle y_{i,1}, \mathbf{y}_{i,2}, ..., \mathbf{y}_{i,T_i} \rangle$, el objetivo consiste en construir

un clasificador h que sea capaz de recibir como entrada una secuencia $\mathbf{x}$, y pueda obtener la secuencia $\mathbf{y}$ mas probable para la misma.

2.4. Enfoques POSTagging

A continuación se describen algunos de los enfoques mas importantes en la solución de POS Tagging:

- Modelo basado en reglas. El modelo basado en reglas es un enfoque de POS tagging que permite la categorización de las palabras usando reglas de reescritura. El tagger de Brill [12] implementa este modelo, en el cual un conjunto de reglas es definido para la anotación de la categoría léxica de las palabras. Estas reglas también permiten el tratamiento de la ambigüedad de las palabras. Un ejemplo de este tipo de reglas es que un artículo puede preceder a un sustantivo y no a un verbo, como por ejemplo con el artículo "*los*", se escribe correctamente en la oración: "los carros", y estaría mal escrito: "los caminar". Otro etiquetador existente, bajo este enfoque es SPOST [16], el cual proporciona niveles altos de presición, en la tarea del POS Tagging para el español.

- Modelo basado en transformaciones.

  Modelo utilizado para el proceso de POS Tagging, en el cual se hace uso de un diccionario y de un corpus previamente anotado (es decir cada palabra con su correspondiente etiqueta). Es un modelo basado en machine learning. Inicialmente se realiza un etiquetado de cada palabra del corpus de entrenamiento haciendo uso del diccionario. Posteriormente se compara dicho etiquetamiento con el etiquetado inicial del corpus, obteniendo una tasa de error. Luego se vuelve a etiquetar ese corpus y se vuelve a comparar con el inicial, de esta forma, intentando obtener una tasa de error menor. Por cada iteración se va obteniendo una regla de transformación. Al final, se obtiene un conjunto de reglas de transformación cuando ya no se pueda reducir más la tasa de error, y es el momento cuando se aplican estas reglas de transformación al corpus sin anotar [13].

- Modelo basado en máxima entropía

  Más que un modelo es una técnica que optimiza el uso de sistemas basados en reglas o HMMs, valiéndose de la asignación de la mayor probabilidad entre las posibles secuencias de categorías. Al hablar de una mayor uniformidad se hace referencia a la máxima entropía que entre ellas pueda ocurrir, es decir, si se tienen todas las palabras etiquetadas con una misma etiqueta, por ejemplo "Verbo", ahí hay baja entropía y por tanto es una cadena a la cual se le asigna baja probabilidad de ser escogida. Posteriormente se entrena el modelo con ejemplos reales, y se calcula las frecuencias de todas las ocurrencias [palabra, etiqueta]. Una vez entrenado el modelo, es utilizado para para el etiquetado de conjutos de palabras.

- Modelo basado en Redes Neuronales. Es un modelo que hace uso de las redes neuronales para solucionar al problema de la ambiguedad, presente en el POS Tagging. Como se observa en [18], se trabajó con el corpus Penn Treebank, el cual está para el idioma inglés, y está basado en el uso del Perceptrón Multicapa y para el entrenamiento, se hace uso del algoritmo Back Propagation. Se entrena el modelo mediante el algoritmo Backpropagation, que propaga el error hacia atrás en la red. En [39] también se trabaja bajo el mismo enfoque.

- Modelo basado en Árboles de Decisión. Modelo que utiliza, para atacar el problema del POS Tagging, árboles binarios de decisión, en [40] se observa un estudio realizado en donde se compara su rendimiento, con el del modelo de HMMs.

2.5. Modelos Gráficos

Los modelos gráficos son utilizados para representar modelos probabilísticos. Algunas de las ventajas mas importantes de los modelos gráficos, es que proveen una manera simple de analizar modelos probabilísticos complejos, además que los cálculos complejos se los puede manipular gráficamente [10].

Un modelo gráfico G comprende un conjunto de nodos V y unas aristas E que conectan dichos nodos. Los nodos representan variables randómicas y los vértices

son las relaciones probabilísticas entre dichos nodos. Si no existe una arista entre dos nodos, se presenta independencia condicional entre los mismos, dado un tercer nodo.

Los modelos gráficos pueden dividirse en dirigidos y no dirigidos. Los modelos dirigidos, también llamados redes bayesianas, en los cuales cada relación entre dos nodos tiene una dirección. Los modelos no dirigidos, son aquellos en donde las aristas entre los nodos no tienen direcciones algunas. Dentro de estos modelos se encuentran los Conditional Random Fields.

2.5.1. Modelos Gráficos Dirigidos

Son aquellos modelos en donde las aristas tienen una dirección, mediante la cual se expresa una relación entre las variables randómicas. La distribución conjunta $p(v)$, se la obtiene a partir del producto de las distribuciones condicionales de cada nodo v_k, donde cada una de esas distribuciones está condicionada a los vecinos de v_k, es decir de aquellos nodos de los cuales llegue una arista hacia el nodo en cuestión v_k^p. [36].

$$p(\vec{v}) = \prod_K = p(v_k|v_k^p)$$

2.5.2. Modelos Gráficos no Dirigidos

Una función potencial Ψ_i es aquella que representa un valor para un clique máximo, en el cual todas las variables randómicas están totalmente conectadas por aristas. La distribución de probabilidad se obtiene por la factorización de funciones Ψ_i, no negativas, de todos los cliques máximos de G [36].

$$p(\vec{v}) = \frac{1}{Z} \prod_C = \Psi_C(\vec{v}_C)$$

Donde Z es una constante que permite la normalización que se define a continuación:

$$Z = \sum_{\vec{v}} \prod_C \Psi_C(\vec{v}_C)$$

	Sol	Lluvia	viento
Sol	0.4	0.5	0.1
Lluvia	0.3	0.4	0.3
Viento	0.3	0.1	0.6

Cuadro 2.1: Matriz de transición para el estado del tiempo

2.6. Cadenas de Markov

Las Cadenas de Markov permiten modelar el comportamiento de procesos estocásticos, es decir no determinísticos, y se fundamentan en encontrar el valor de un estado, basado en el estado inmediantamente anterior y la probabilidad de cambiar de ese estado anterior hacia el que se desea encontrar. Una cadena de Markov finita tiene un número de estados finito. Se definen 3 elementos principales: estado, transición y probabilidad condicional. El estado hace referencia a los valores que toma un conjunto de variables en un determinado instante de tiempo. La transición es el cambio de un estado a otro estado. Al ser el proceso estocástico, no se tienen los valores de los estados en todos los instantes de tiempo, sino la probabilidad de cambio de un estado a otro, dependiendo del estado anterior, este concepto se denomina probabilidad condicional [37].

La probabilidad de pasar de un estado s_i a un estado s_j se la obtiene mediante p_{ij}. Donde i hace referencia a la fila de la matriz de transición y j hace referencia a la columna en la misma matriz.

En la figura 2.2, se observa la gráfica de un modelo correspondiente a tres estados del tiempo que pueden presentarse en un determinado día. La matriz de transición que se observa en la tabla 2.1 hace referencia a la distribución de probabilidad para dicho modelo del tiempo, en donde se observa los estados del tiempo del día de hoy y la probabilidad del estado del tiempo para el día de mañana. Cada fila indica el estado del tiempo del día de hoy y las columnas indican el probable estado del tiempo del día de mañana, donde las celdas indican la probabilidad de dicho estado futuro. De esta forma suponiendo que el estado del día de hoy i= sol, la probabilidad p_{ij} que el día de mañana el estado j= lluvia es de 50 %.

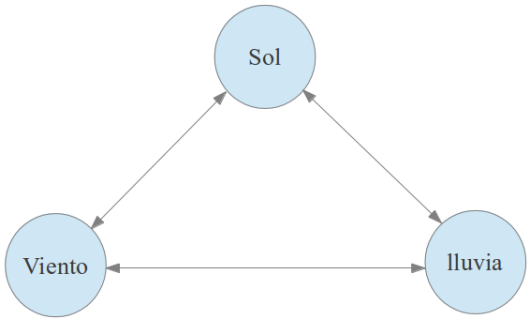


Figura 2.2: Ejemplo estados del tiempo

2.6.1. N-Gramas

Introduciendo las cadenas de markov en el PLN, se puede realizar la predicción de una palabra, dadas las anteriores palabras a ella, en una determinada oración. Los modelos que permiten realizar esta tarea se denominan N-Gramas. Para ejemplo se tiene la siguiente oración h: 'El café colombiano es '. Se desea encontrar la probabilidad que la próxima palabra es 'delicioso'. Utilizando conteos de frecuencia relativa se obtiene la probabilidad buscada: P(delicioso | El café colombiano es)= C(El café colombiano es delicioso)/C(El café colombiano es). Para efectos de eficiencia se realiza el conteo de palabras, y estimar su probabilidad, dando lugar a los N-Gramas, en los cuales se predice la probabilidad de la siguiente palabra, dados las N-1 anteriores palabras. Cuando se desea predecir la siguiente palabra en una oración determinada, se utilizan los bigramas, mediante los cuales, es necesario únicamente la palabra anterior. En este sentido, el cálculo de dicha probabilidad está dado por

$$P(w_{i-1}, w_i) = P(w_i|w_{i-1})$$

Esto es, para calcular la probabilidad de la palabra 'delicioso', en el mencionado ejemplo, se debe calcular P(delicioso | es) = C(es delicioso)/C(es). La suposición de Markov dicta que, una palabra solo depende de la anterior palabra. Haciendo el uso de bigramas, se observa de la siguiente forma:

$$P(w_n|w_{n-1}, w_{n-2}, \ldots w_1) = P(w_n|w_{n-1})$$

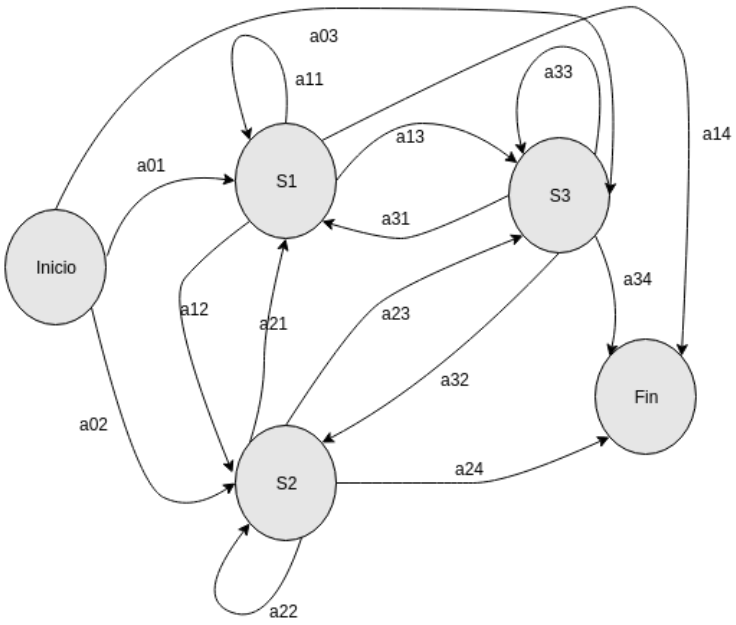


Figura 2.3: Ejemplo automata HMM

2.7. Cadenas Ocultas de Markov

Un HMM es un modelo estocástico, que obedece a un proceso de markov pero con parámetros desconocidos. En una cadena oculta de markov, los elementos observables corresponden a los simbolos de salida, los cuales son generados por un conjunto de estados ocultos, bajo una probabilidad de emisión. De ahí que las HMMs corresponden a un modelo generativo.

Las HMMs se define por los siguientes componentes [25]:

$S = \{S_1, S_2, ..., S_n\}$ es un conjunto que contiene el alfabeto de estados.

$V = \{V_1, V_2, ..., V_m\}$ hace referencia a un conjunto de simbolos observables.

$A = \{a_{ij}\}$ Que hace referencia a la matriz de probabilidades de moverse desde un estado i en un tiempo t, hasta un estado j, en un tiempo $t + 1$, donde i toma los valores desde 1 hasta n y j toma los valores desde 1 hasta n.

$B = \{b_j(k)\}$, que corresponde a la probabilidad de observar el simbolo V_k generado

por el estado S_j.

$\pi = \{\pi_i\}$ Que corresponde a una distribución inicial, donde el estado en un tiempo t_1 es S_i, donde $1 \leq i \leq n$.

El modelo, con los coponentes mencionados puede generar secuencias de elementos observables tomados de V.

La matriz A es generada a partir de un autómata, el cual contiene todos los estados y relaciones entre estos, que representan las probabilidades de transición desde un estado hacia otro. En la figura 2.3, se observa un ejemplo de un autómata con 3 estados.

En la figura 2.4, se observa la generación de la matriz B, donde se computa la probabilidad de que un elemento observable V_k sea generado a partir de un estado S_j, donde $(1 \leqslant k \leqslant \mathrm{m})$ y $(1 \leqslant j \leqslant \mathrm{n})$.

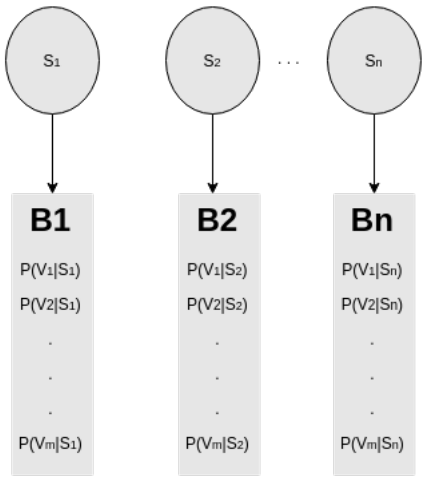


Figura 2.4: Generación matriz B

2.7.1. Suposiciones que hace las HMMs

En las Cadenas Ocultas de Markov se hacen las siguientes suposiciones. Primero se asume que cada estado depende únicamente del estado inmediatamente ante-

rior, esto es, cada estado y_t, depende de y_{t-1}, siendo independiente de los estados $y_{t-2}, ..., y_2, y_1$. En este orden de ideas se puede calcular la probabilidad $p(y_t)$ dado $p(y_{t-1})$, en otras palabras $p(y_t|y_{t-1})$. Segundo, las HMMs, asumen que cada observación x_t, depende únicamente del estado y_t. De igual manera se puede calcular la probabilidad $p(x_t)$, dado $p(y_t)$. Con el cálculo de las anteriores probabilidades, se puede obtener la probabilidad conjunta $p(y|x)$.

Dada una matriz de probabilidad de transición A, una secuencia de probabilidades de emisión B, un alfabeto S de tamaño n y una secuencia de observaciones X de tamaño T, existe un total de (T^n) posibles secuencias Y, de estados ocultos. Para encontrar la secuencia con máxima probabilidad $argmaxp(y|x)$ se utiliza el algoritmo de Viterbi.

2.7.2. Algoritmo de Viterbi

Es un algoritmo de programación dinámica cuyo objetivo es conseguir la secuencia de etiquetas cuya probabilidad es máxima, dada una secuencia de observaciones O, una matriz de probabilidades de transición entre estados A y una matriz de probabilidad de emisiones de símbolos desde los estados ocultos B.

Inicialmente se construye una matriz, donde la cantidad de filas corresponde con la cantidad de estados que se dispone en el alfabeto de estados, incluyendo dos filas adicionales para el estado de inicio y el estado final. El número de columnas corresponde con el número de observables mas una columna adicional, para el estado inicial, dentro de la secuencia O=$o_1 o_2 .. o_T$. Cada celda de dicha matriz indica la probabilidad de encontrarse en un estado s_j, en un tiempo t, donde t corresponde con cada columna de la matriz, dada una observación o_t.

El algoritmo es recursivo en el sentido que, para encontrar la probabilidad de un estado s_j en un tiempo t, se tiene que haber calculado la ruta mas probable para llegar a él, la cual corresponde con una secuencia de estados compuesta por

$s_0 s_1 ... s_{t-1}$.

Formalmente

$$e(j)=max_{i=1}^{n}\ e_{t-1}(i)a_{ij}b_j o_t$$

Donde

- $e_{t-1}(i)$ corresponde el valor máximo de la probabilidad de la secuencia de estados $s_1 s_2 .. s_{t-1}$.
- a_{ij} hace referencia a la probabilidad de transición del estado actual s_j dado el estado previo s_i.
- $b_j o_t$ corresponde a la probabilidad de que se emita el símbolo o en dado el estado s_j en el tiempo t.

2.7.3. HMMs en POStagging

Aplicando las cadenas ocultas de Markov en Part-of-Speech Tagging, se tiene que encontrar la secuencia de etiquetas cuya probabilidad sea la máxima entre todas las posibles combinaciones. Para esto se tiene en cuenta la probabilidad de que se presente una palabra dada una etiqueta en particular, y este valor se lo multiplica por la probabilidad de que se presente una etiqueta, dada su etiqueta anterior.

En la figura 2.5 se aprecian los nodos observables, es decir la sucesión de palabras que forman la oración "Secretariat is expected to race tomorrow", ejemplo tomado de [23]. Se presenta ambigüedad en la palabra "*race*". Para resolver este problema se hace uso del 87-tag Brown Corpus tagset.

Para la secuencia de etiquetas (b) se obtiene la probabilidad $P(NN|TO) =$ 0,00047, es decir de que la etiqueta NN se presente dado que la anterior etiqueta sea TO. Se calcula posteriormente la probabilidad $P(race|NN) =$ 0,00057, es decir,

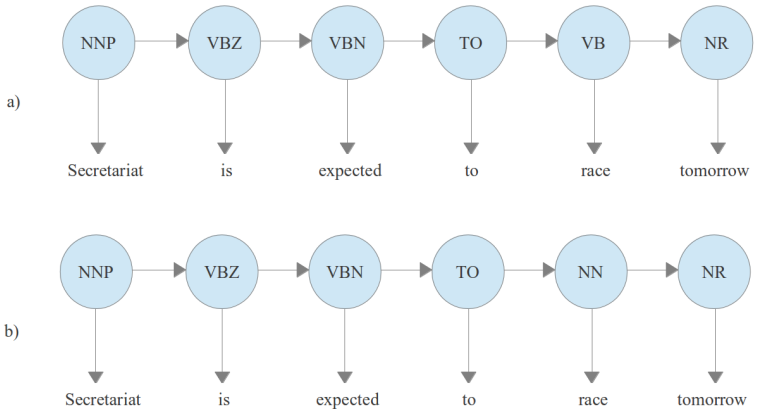


Figura 2.5: Ejemplo ambigüedad palabra race, tomado de [23]

según el nombrado corpus, la palabra "race" cuantas veces corresponde a una etiqueta NN. Finalmente se calcula la probabilidad $P(NR|NN) = 0{,}0012$, al multiplicar estos valores se obtiene el resultado 0.00000000032.

El mismo procedimiento se realiza con la secuencia (a), obteniendo el resultado 0.00000027. Finalmente se escoge la que presenta mayor probabilidad, siendo la secuencia de etiquetas representada en (a).

2.8. Modelos de Markov de Máxima Entropía

Denominado MEMM (por sus siglas en inglés), Es un modelo gráfico, discriminativo, basado en cadenas de Markov, utilizado para el etiquetado secuencial. Es soportado en el modelo de máxima entropía. Los estados en el modelo están conectados en un grafo dirigido, como se observa en la figura 2.6. Dado que es un modelo discriminativo, permite la inclusión de características para las observaciones. La importancia de estos modelo radica en que ellos calculan la probabilidad condicional de manera directa mediante el uso de funciones características. La probabilidad de un estado depende del estado anterior y de la observación para él mismo. La probabilidad de una etiqueta en un estado, dado el estado anterior y la observación para

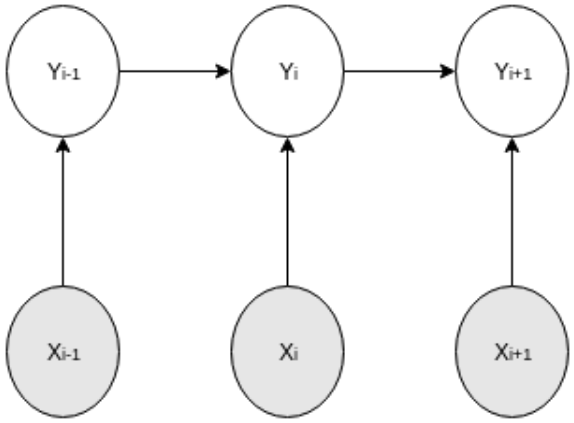


Figura 2.6: Estructura gráfica MEMM

el mismo, está dada por:

$$p(q|q', o) = \frac{1}{Z(o,q')} \exp \{\sum_i w_i f_i(o, q)\}$$

La probabilidad de una secuencia de etiquetas es como se observa a continuación:

$$p(Q|O) = \prod_{i=1}^{n} P(q_i|q_{i-1}, o_i)$$

En donde $P(q_i|q_{i-1}, o_i)$, indica la probabilidad presentada por una transición entre el estado q_{i-1} y el estado q_i, dada la observación o_i [28]. Donde $f_i(o, q)$, es una función característica que indica alguna propiedad sobre la observación o el estado. W_i indica el peso que tiene dicha f_i. Z es un factor de normalización que permite que la distribución sume 1. El modelo MEMM tiene implícito el problema del sesgo de la etiqueta [24], en donde las transiciones que salen de un estado a otro, no compiten con todas las transiciones en el modelo, sino con algunas.

2.9. Campos Randómicos Condicionales

Es un modelo gráfico, no dirigido, el cual hace referencia a un Markov Random Field, en el cual las salidas "Y" son globalmente, condicionadas con un conjunto de observaciones "X", como se puede observar en la figura 2.7.

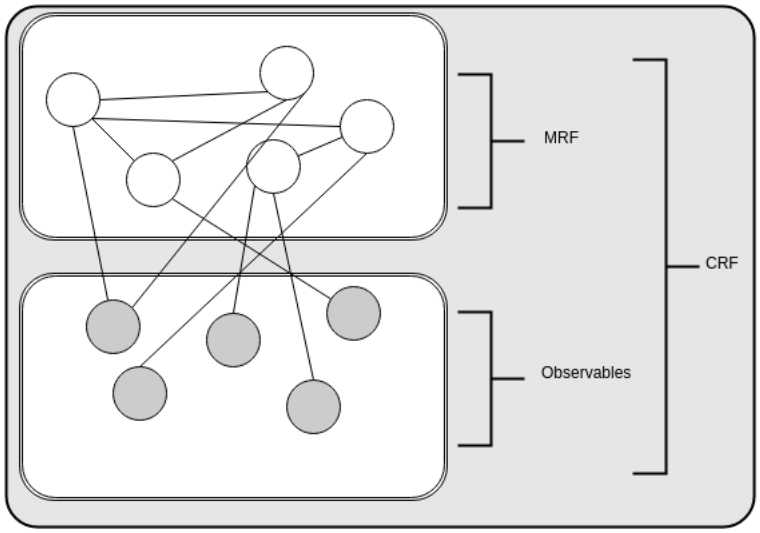


Figura 2.7: CRF como resultado de $Y \cup X$. Donde Y corresponde al MRF y X son los observables.

Sea $G = (V, E)$ un grafo, tal que $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$, de tal forma que $\mathbf{Y}$ es indexado por vertices de G. Entonces $(\mathbf{X}, \mathbf{Y})$ es un CRF si $\mathbf{Y}_v$ obedece la propiedad de Markov, esto es: $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$ donde $w \sim v$ significa que w y v son vecinos en el grafo G, estando condicionado en $\mathbf{X}$ [24].

En otras palabras, los estados ocultos Y, están unidos mediante aristas de E y además cumplen con la propiedad de Markov, en cuanto a que la probabilidad de Y_i dados todos los estados de Y que no sean $_i$, $p(Y_i|Y_{j \neq i})$ es igual a la probabilidad de Y dados sus vecinos $p(Y_i|Vecinos(Y_i))$, es decir, todas aquellas variables no observables que tienen arista directa con Y, esto implica que, una variable es condicionalmente independiente de todas las demás variables, dados sus vecinos, convirtiéndose en un MRF. Estos estados están condicionados en X, el cual corresponde a un conjunto de observaciones, lo cual lo convierte en un CRF, como se observa en la figura 2.7.

De esta forma [47] define un Conditional Random Field (CRF) como una 6-tupla:

$$M = (L, \alpha, Y, X, \Omega, G)$$

Donde

L es un conjunto que contiene el alfabeto de etiquetas de salida.

α es un conjunto que contiene el alfabeto de elementos de entrada, u observables.

Y es un conjunto de variables, no observables, el cual toma valores de L.

X es un conjunto fjo, de variables observables, el cual toma valores de α.

Ω es un conjunto de funciones potenciales, $\phi_c = (\mathbf{y}, \mathbf{x})$ en donde $\{\phi_c : L^{|Y|} \times \alpha^{|X|} \rightarrow \mathbb{R}\}$.

$\phi_c(\mathbf{y}, \mathbf{x})$ Únicamente depende del grupo de variables no observables que están unidas mediante aristas $c \subseteq Y$.

Una función potencial no se la puede interpretar probabilísticamente, sino que representa un peso sobre las variables para las cuales está definida. Cada función potencial opera solo sobre un conjunto de variables ramdómicas, las cuales están unidas por aristas, formando un clique máximo. Teniendo en cuenta la estructura gráfica del CRF, se factoriza la distribución conjunta sobre los elementos Yc de Y, en un producto, el cual es normalizado, de funciones potenciales cuyos valores son reales y positivos.

La probabilidad condicional Y, teniendo en cuenta los observables X, y Ψ_A corresponde al grupo de factors en G, es $P(\mathbf{y}|\mathbf{x})$ [5].

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{\Psi_A \in G} \exp \{ \sum_{k=1}^{K(A)} \lambda_{Ak} f_{Ak}(y_A, x_A)\}$$

Donde Z una constante, la cual es denominada función de partición. La cual es obtenida a través de la suma de todas las posibles secuencias de los estados. Y su objetivo es lograr la normalización.

$$Z = \sum_{\mathbf{y'}} \prod_{\Psi_A \in G} \exp \{ \sum_{k=1}^{K(A)} \lambda_{Ak} f_{Ak}(\mathbf{y'}_A, x_A)\}$$

2.9.1. CRF de Cadena Lineal

Corresponde a una topología de CRF dónde los estados ocultos forman una secuencia lineal, tal y como se observa en la figura 2.8.

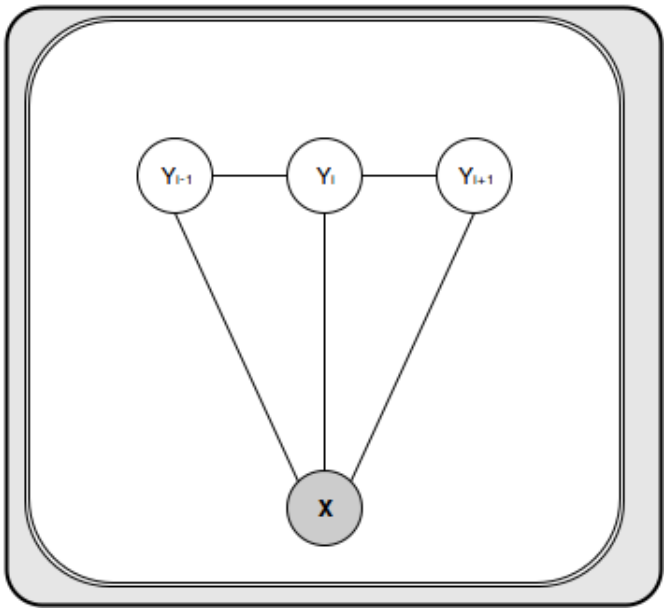


Figura 2.8: Linear Chain CRF, donde $Y_{i-1}, Y_{i-1}, Y_{i+1}$ corresponden a los estados ocultos y $\mathbf{X}$ corresponde a todos los elementos observables

En el caso del CRF de cadena lineal, las funciones potenciales operan sobre pares adyacentes de vertices de los estados ocultos Y, por lo tanto la probabilidad de distribución condicional $p(\mathbf{y}|\mathbf{x})$, queda de la siguiente forma:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z} \prod_{i=1}^{n} \Psi_i(y_i, y_{i-1}, \mathbf{x})$$

Donde Z(X) corresponde a la función de partición.

$$Z = \sum_{\mathbf{y'}} \prod_{i=1}^{n} \Psi_i(\mathbf{y'}_i, \mathbf{y'}_{i-1}, \mathbf{x})$$

Y la función potencial Ψ_i, es definida como:

$$\Psi_i(y_i, y_{i-1}, \mathbf{x}) = \exp\{\sum_{k=1}^{K} \lambda_k f_k(y_i, y_{i-1}, \mathbf{x})\}$$

Las funciones $f_k(y_i, y_{i-1}, \mathbf{x})$ se denominan funciones características, y están asociadas a las transiciones entre los estados y_i, y_{i-1} y todo el conjunto de observables $\mathbf{x}$, todo visto desde el estado i.

Las funciones características tienen el objetivo de servir como indicador de una cierta propiedad y toma valores de 0 y 1. En estas funiones se puede llegar a utilizar cualquier tipo de información de los observables X.

Algunos ejemplos de funciones características son como siguen:

$$f_k(y_i, y_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } y_{i-1} = \text{verbo e } y_i = \text{sustantivo} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

Función característica que tiene el valor de 1, en caso que el estado actual es sustantivo y el estado anterior es un verbo. En caso contrario toma el valor de 0.

$$f_k(y_i, y_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } y_i = \text{verbo y } x_{i-1} = \text{'el'} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

Función característica que tiene el valor de 1, en caso que el estado actual es verbo y la palabra anterior es 'el'. En caso contrario toma el valor de 0.

$$f_k(y_i, y_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } y_i = \text{verbo y } x_{i+1} \text{ es de género masculino} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

Función característica que tiene el valor de 1, en caso que el estado actual es verbo y la palabra siguiente tiene género masculino. En caso contrario toma el valor de 0.

El conjunto $\{\lambda_k\} \in \Re^K$ corresponde a los parámetros del modelo los cuales deben ser estimados mediante entrenamiento.

2.9.2. Entrenamiento

Se cuenta con un conjunto de datos de entrenamiento $(\mathbf{x}^{(k)}, \mathbf{y}^{(k)})$, donde $x^{(k)} \in$X, $y^{(k)} \in$ y, los cuales son independientes e idénticamente distribuidos. Para este proceso se utiliza la máxima verosimilitud regularizada. Se trata de encontrar los parámetros λ que maximicen la función de verosimilitud, dada por [46]

$$\ell(\lambda) = \sum_k \left[log \frac{1}{Z(x^{(k)})} + \sum_j \lambda_j F_j(\mathbf{y}^{(k)}, \mathbf{x}^{(k)}) \right]$$

La función es cóncava y converge hacia el máximo global. Como una medida para evitar el sobreentranamiento se utiliza la regularización, mediante la cual se penaliza a aquellos valores demasiado grandes. Se utiliza para esto, la distancia de Euclidean y un parámetro de regularización [5].

Al derivar esta función con respecto a los parámetros λ_j se obtiene [46]

$$\frac{\partial \ell(\lambda)}{\partial \lambda_j} = E_{p(\mathbf{Y},\mathbf{X})} \left[F_j(\mathbf{Y}, \mathbf{X}) \right] - \sum_k E_{p(\mathbf{Y}|\mathbf{x}^{(k)},\lambda)} \left[F_j(\mathbf{Y}, \mathbf{x}^{(k)}) \right]$$

Donde el primer término hace referencia a los datos de entrenamiento y el segundo término lo que se espera con respecto a la distribución p.

Métodos como algoritmos de escala iterativa, conjugación del gradiente, LBFGS son utilizados para la realización de la optimización [5]. El algoritmo LBFGS [21] es muy utilizado, debido a que calcula una aproximación a la Matriz Hessiana, con memoria limitada.

2.9.3. Inferencia

En CRF, al igual que en los HMMs se utiliza para la inferencia los algoritmos de Forward, Backward y Viterbi. Forward es un algoritmo recursivo, que utiliza un vector de variables cuyo tamaño es el mismo que la cantidad de estados. Se define por [38]

$$p(y_j = y, y_{j-1} = y'|\mathbf{x}) = \frac{\Psi_j(y,y',\mathbf{x}).\alpha_{j-1}(y')}{Z(\mathbf{x})}$$

Donde, para todas las variables Forward

$$\alpha_j(y) = \sum_{y'} \Psi_j(y, y', \mathbf{x}).\alpha_{j-1}(y')$$

El cálculo de las variables Backward se lo obtiene de manera similar, pero tomando en cuenta el tiempo j+1 [38].

$$p(y_j = y, y_{j-1} = y'|\mathbf{x}) = \frac{\Psi_j(y,y',\mathbf{x}).\beta_j(y)}{Z(\mathbf{x})}$$

Donde, para todas las variables Backward

$$\beta_j(y) = \sum_{y'} \Psi_{j+1}(y', y, \mathbf{x}).\beta_{j+1}(y')$$

El algoritmo de Viterbi es un algoritmo de programación dinámica que permite encontrar la secuencia Y que haga que se maximice la distribución de probabilidad condicional, dado X y dado el modelo el cual está entrenado, es decir, los parámetros que fueron calculados. El algoritmo se basa en resultados anteriores para encontrar los actuales. Al final el algoritmo hace backtracking para encontrar la secuencia mas probable. Está dado por [38]

$$\mathbf{y}^* = argmax_y p(\mathbf{y}|\mathbf{x})$$

$$v_j(y) = argmax'_y \Psi_j(y, y', \mathbf{x}).v_{j-1}(y')$$

Donde $v_j(y)$ almacena el valor correspondiente a la mas probable secuencia Y, de tamaño j, cuya etiqueta final es y.

2.10. Lingüística de Corpus

Un corpus hace referencia a una colección de elementos del lenguaje que son almacenados en medios informáticos y que son utilizados para estudios lingüísticos. Son tomados de fuentes originales de lenguaje escrito y hablado, los cuales han sido anotados por lingüístas, expertos en esta área [1]. Las areas que están trabajadas en un corpus son: Análisis morfológico, Análisis Léxico, Sintáxis, Análisis discursivo, Fonología, entre otras .

En [2] se clasifican a los corpus según el proceso al que se someten, y pueden ser simples, los cuales no tienen ningún tipo de anotación, verticales, los cuales son listados de forma vertical y tienen frecuencias frente a cada palabra, y los corpus anotados, los cuales tienen anotaciones gramaticales, morfológicas, entre otras.

2.10.1. AnCora

Es un corpus con diferentes niveles de anotación, tanto para el español como para el catalán, desarrollado en el CLiC-UB (Centre de Llenguatge i Computaci'o, Universitat de Barcelona) y en el TALP, Software Department, Universitat Politécnica de Catalunya. Dentro de los niveles de anotación se encuentran [45] [1].

- Lema (raíz) y categoría morfológica
- Constituyentes y funciones sintácticas
- Estructura argumental y papeles temáticos
- Clase semántica verbal

También contiene para el español un léxicon verbal y otro de nominalizaciones deverbales. El corpus AnCora contiene 500000 palabras anotadas multinivel. Con-

[1] información completa en http://clic.ub.edu/corpus/webfm send/13

tiene información anotada para las tareas de análisis morfológico, análisis sintáctico y semántico, entre otras. Para el lenguaje español la información morfológica y sintáctica está cubierta completamente, mientras que las anotaciones semánticas están cercana al 40 % de cubrimiento[1]. El corpus de ancora puede ser descargado libremente desde su sitio web[2].

Para este trabajo se utilizó el corpus AnCora [45], para el cual se utilizó la etiquetación POS de cada palabra, tanto para el entrenamiento como para el testeo del modelo. Dicho corpus utiliza el sistema de anotación [29]. Las etiquetas lingüísticas que éste comprende son:

- Nombre, en donde se puede encontrar el género, el número.
- Verbo, al que se le puede obtener el tiempo, la persona, el número y el género.
- Adjetivo, del cual se puede encontrar el género y el número.
- Pronombre, del cual no se puede extraer ninguna característica de las propuestas en la presente investigación.
- Determinante, del cual se puede extraer la persona, el género y el número.
- Adverbio, del cual no se puede extraer ninguna característica de las propuestas en la presente investigación.
- Preposición, del cual se puede extraer el género y el número.
- Conjunción, del cual no se puede extraer ninguna característica de las propuestas en la presente investigación.
- Numeral, del cual no se puede extraer ninguna característica de las propuestas en la presente investigación.

[2]http://clic.ub.edu/ancora

- Interjección, del cual no se puede extraer ninguna característica de las propuestas en la presente investigación.
- Puntuación, del cual no se puede extraer ninguna característica de las propuestas en la presente investigación.
- Fecha y Hora, del cual no se puede extraer ninguna característica de las propuestas en la presente investigación.

2.11. Estudios y aplicaciones bajo el modelo CRF

A continuación se mencionan algunas aplicaciones para POS Tagging y estudios que se han desarrollado bajo el modelo de CRFs en el idioma español.

CRF ha sido utilizado en algunos estudios, a saber, en [9] se hace uso del tageador CRF++ Tagger aplicado al Assamese, el cual es un lenguaje de la zona nororiental de la India, alcanzando porcentajes no tan altos en la tarea del POS Tagging, debido a que no existe un corpus ampliamente anotado. Igualmente en [33] se hace realiza una investigación utilizando algunos lenguajes de la India, en la que se obtienen resultados accuracy para el Telegú, Hindi y Bengalí, de 77.37 %, 78.66 % y 76.08 %, respectivamente, en la tarea del POS Tagging utilizando CRF y Aprendizaje Basado en Transformaciones. La tarea del POS Tagging ha sido objeto de estudio en [32] en donde se aplica CRF obteniendo on porcentaje de accuracy de 69,40 para el Hindi.

En [43] se hacen esfuerzos por intentar aumentar el rendimiento de la tarea del POS Tagging en el idioma Nepali, el cual es hablado en Nepal, Bután, Birmania y en algunas partes de la India. La tarea fue realizada utilizando un lexicón, con palabras previamente anotadas, luego se aplicó el modelo de HMM y seguidamente se ejecutó un sistema basado en reglas. A pesar que cuentan con la ausencia de grandes corpus y de reglas gramaticales se alcanzó un nivel de precisión de 93.15 %. En [49] se realiza un estudio de la tarea del POS Tagging, igualmente en el idioma nepalí, pero en este caso utilizando solo el modelo HMM y el dataset NELRALEC, alcanzando una precisión de 95.43 %. Para el Nepalí, igualmente se realiza la tarea del POS Tagging, evaluando el modelo de máquinas de soporte vectorial [42], obteniendo una precisión del 96.48 %. En [35] se utiliza el enfoque basado en reglas para la realización del POS Tagging en el idioma indonesio, en donde se utiliza un diccionario de dicha lengua, como recurso lingüístico. Se ejecuta, antes del POS Tagging, la tarea del reconocimiento de entidades, para encontrar sustantivos propios. El porcentaje

alcanzado, en este trabajo es del 79 %. En [22], se documenta la realización de la tarea de POS Tagging en textos arábigos, utilizando maquinas de soporte vectorial lineales, modeladas con características léxicas básicas, alcanzando un porcentaje del 97.15 %, donde uno de sus productos corresponde con un software que pertenece al MADAMIRA software suite. La tarea del POS tagging fue realizada en [15] utilizando el ODIA Corpus, el cual corresponde con un lenguaje de la India, y el oficial en Odisha [8]. En este caso se utilizaron máquinas de soporte vectorial, obteniendo una precisión del 82 %, además que se cuenta con conjunto de etiquetas de tamaño 5. En general en [3] se realiza un estudio de POS Tagging para Lenguajes de la India, siendo utilizado el enfoque basado en reglas, y enfoque estocásticos.

En [14] se investiga a gran nivel de detalle los etiquetadores: IULA TreeTagger, Default TreeTagger, FreeLing, IXA pipes, los cuales realizan la tarea de POS Tagging para el español, donde concluyen que con IULA obtienen el mejor rendimiento. Aunque en dicho estudio se centran en las características de la herramienta POS Tagger, en cuanto a cómo tratan y su soporte que prestan a aquellos elementos del idioma español como por ejemplo lemas, clíticos verbales, entre otros, mas que en los modelos que los soportan, además de que no tienen en cuenta el entrenamiento de los mismos.

Y aunque en [17] se intenta demostrar que la tarea del POS tagging no está resuelta en un porcentaje del 100, para ello hacen uso de un corpus alemán y de 5 herramientas POS tagger. Hay que tener en cuenta la cantidad de esfuerzos que se han hecho para que cada cada vez se perfeccionen las técnicas en cuanto a POS Tagging se refieren.

Pero los CRFs no han sido utilizados únicamente para la tarea del POS Tagging, en [34] se realiza un estudio para conocer la opinión de un grupo de usuarios acerca de un determinado producto, si ésta es positiva, negativa o neutral, utilizando los CRFs aplicado a los comentarios que los usuarios de dicho producto escriben vía

web. Se realiza una comparación del uso de CRFs en este contexto, con el uso de HMMs, teniendo CRFs un mayor rendimiendo.

En cuanto a aplicaciones que son basadas en el Modelo de CRFs se tienen las siguintes:

- CRF++ es un software de código abierto que implementa el modelo de CRFs, diseñado para propósito general. Utilizado para la segmentación y el etiquetado de los datos, es desarrollado bajo la licencia GNU Lesser General Public License[3].

- CRFTagger[4], es un tagger para el inglés, desarrollado bajo el modelo de CRFs, sobre en java, basado en el tagger flexCRFs. Ofrece una presición del 97 %. Está desarrollado bajo licencia GNU General Public License version 2.0.

- FlexCRFs[5], herramienta desarrollada en lenguaje C++ bajo el modelo de CRFs, bajo licencia GNU General Public License.

- Carafe [6], es otra herramienta software que está basada en CRFs, desarrollada en java bajo la licencia BSD.

- Stanford NER es una herramienta que implementa CRFs, y que está desarrollada bajo java con licencia GNU General Public License version 2.0[7], utilizada para el reconocimiento del nombre de las entidades.

[3]disponible en http://crfpp.googlecode.com/svn/trunk/doc/index.html
[4]disponible en http://crftagger.sourceforge.net/
[5]disponible en http://www.jaist.ac.jp/hieuxuan/flexcrfs/flexcrfs.html
[6]disponible en http://sourceforge.net/projects/carafe
[7]disponible en http://nlp.stanford.edu/software/CRF-NER.shtml

3. Clasificación de las categorías utilizando CRFs

En esta parte del trabajo se propone la solución del problema del POS Tagging como una tarea de etiquetado secuencial utilizando aprendizaje automático. Por esta razón se propone un clasificador, basado en aprendizaje supervisado, de categorías linguísticas, para sentencias en el español. En este capítulo se habla del desarrollo del clasificador, la selección de características, la puesta en marcha y los algoritmos de extracción. Igualmente se resalta la importancia de usar CRFs como modelo de anotación secuencial.

3.1. Preprocesamiento

En la figura 3.1 se observa que esta etapa incluye dos tareas. Tokenización, en la que se segmenta la sentencia de entrada en cada una de sus palabras. Y extracción de características morfológicas, en donde se extraen las características de cada una de las palabras, de cada una de las sentencias que recibe como entrada. De cada palabra se extraen las siguientes características: número, persona, tiempo, género. Además de esas características, para cada palabra se tienen características locales a la palabra, como son: la palabra comienza con letra mayúscula, la palabra contiene, al menos, un símbolo, la palabra contiene algún dígito, el tamaño de la palabra, si es superior a 3 letras o no.

En la tabla 3.1, se presentan todas las características utilizadas, tanto en el entrenamiento como en el testeo del clasificador. Donde i hace referencia a la posición actual de la palabra.

La selección de las características obedece a su estructura morfológica y sus pro-

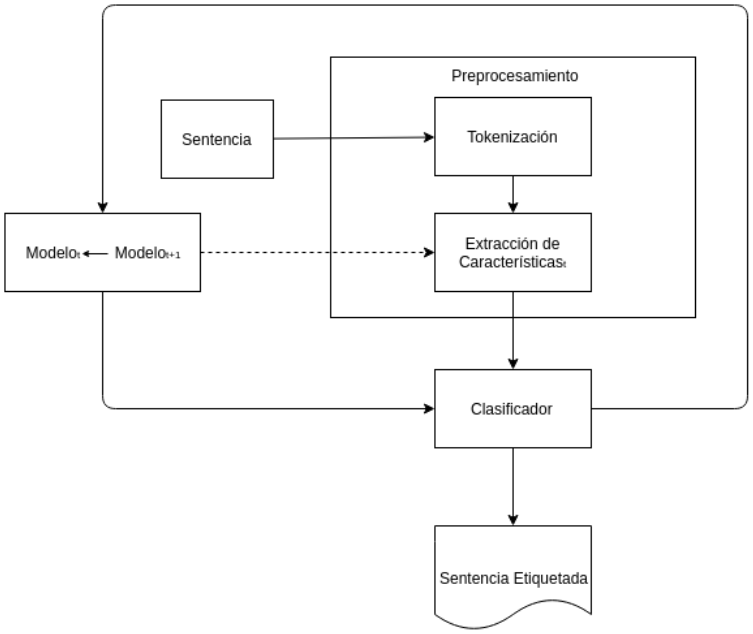


Figura 3.1: Arquitectura del clasificador

piedades locales. En el caso de la característica tamaño de la palabra, toma el valor de 1, cuando el tamaño de la palabra es superior a 3 caracteres, caso contrario toma el valor de 0. Para la característica la palabra comienza con mayúscula, toma el valor de 1 cuando la palabra comienza con letra mayúscula, caso contrario toma el valor de 0. Para el caso de la característica la palabra contiene digitos, toma el valor 1 cuando la palabra contiene algún dígito y la característica la palabra contiene símbolos, toma 1 cuando la palabra contiene algún símbolo, que no corresponda a letra o a dígito.

El preprocesamiento depende del modelo t, como se visualiza en la figura 3.1 utilizado, donde $1 \leqslant t \leqslant 18$. es decir, cada modelo t tiene una combinación diferente de características morfológicas y locales a la palabra.

Al finalizar esta etapa se obtiene el conjunto preprocesado, en donde cada una de las sentencias cumple las características anteriormente descritas.

Una vez obtenido el conjunto preprocesado, este alimenta el algoritmo que se

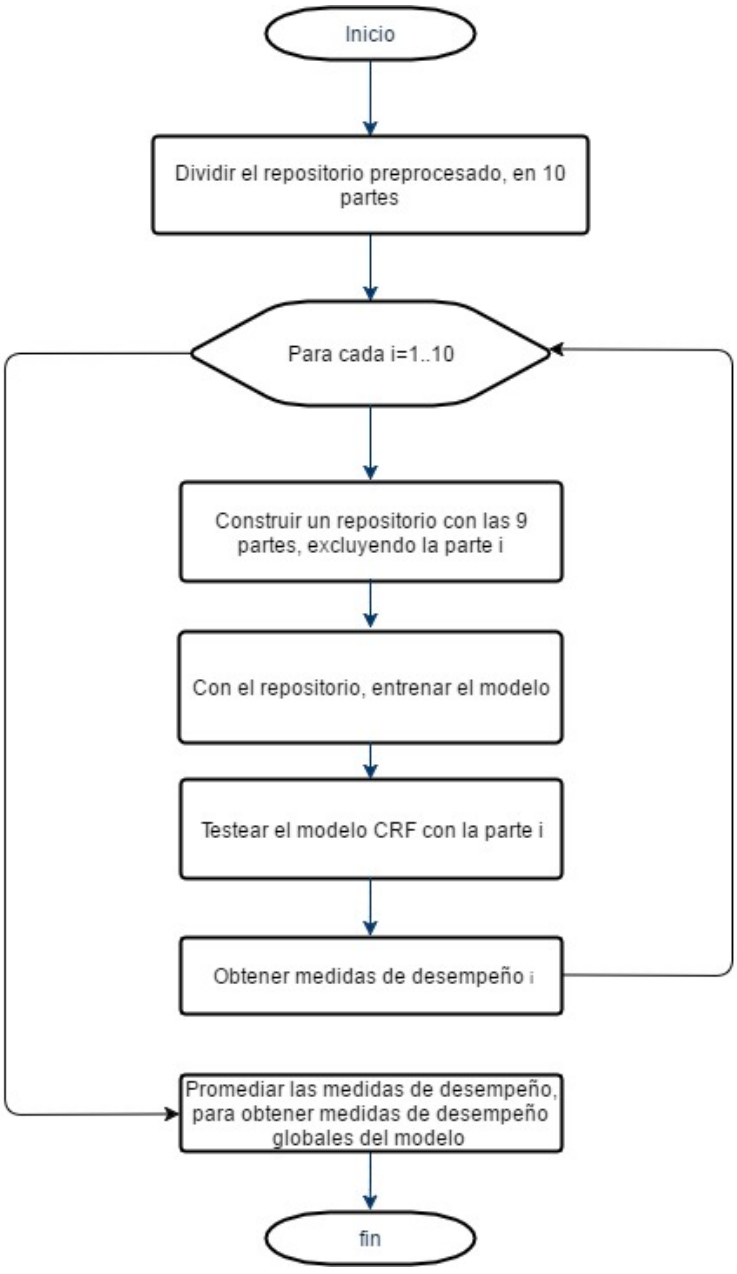


Figura 3.2: Algoritmo aplicación de POS Tagging con el Modelo CRF

Cuadro 3.1: Características seleccionadas

Etiqueta en la plantilla	Descripción de la característica
o_i	El token de la sentencia.
g_i	El género de la palabra. $g_i \in$ { Masculino, Femenino, 0 }
n_i	El número de la palabra. $n_i \in$ { Singular, Plural, 0 }
p_i	La persona de la palabra. $p_i \in$ { Primera, Segunda, Tercera , 0 }
t_i	El tiempo de la palabra. $t_i \in$ { Presente, Pasado, Futuro, Imperfecto, Condicional, 0 }
l_i	El tamaño de la palabra. $l_i \in$ { 1, 0 }
m_i	La palabra comienza con mayúscula. $m_i \in$ { 1, 0 }
d_i	La palabra contiene dígitos $d_i \in$ { 1, 0 }
s_i	La palabra contiene símbolos $s_i \in$ { 1, 0 }

visualiza en la figura 3.2, mediante el cual se divide el conjunto en 10 partes, luego se crean conjuntos conformados por 9 partes, para entrenar el modelo. El modelo, una vez entrenado, es testeado con la parte restante. Esto se realiza 10 veces, teniendo en cuenta la validación cruzada que se utilizó en el presente trabajo. De esta forma, para cada parte $_i$ se obtienen las medidas de desempeño, las cuales, al final son promediadas para obtener el rendimiento general del modelo.

3.2. Clasificación de categorías mediante el etiquetado secuencial

La tarea de POS Tagging para el español puede ser propuesta como un problema de etiquetado secuencial mediante el uso de los CRFs, en el cual, dada una secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$ con su correspondiente secuencia de géneros $\mathbf{g} = (g_1, g_2, ...g_r)$, su correspondiente secuencia de números $\mathbf{n} = (n_1, n_2, ...n_r)$, su correspondiente secuencia de personas $\mathbf{p} = (p_1, p_2, ...p_r)$, su correspondiente secuencia de tiempos $\mathbf{t} = (t_1, t_2, ...t_r)$, su correspondiente secuencia de tamaños $\mathbf{l} = (l_1, l_2, ...l_r)$, su correspondiente secuencia de tipos de comienzo de palabra $\mathbf{m} = (m_1, m_2, ...m_r)$, su correspondiente secuencia de indicaciones de existencia de dígitos $\mathbf{d} = (d_1, d_2, ...d_r)$ y su correspondiente secuencia de indicaciones de

existencia de símbolos $\mathbf{s} = (s_1, s_2, ...s_r)$, el objetivo es encontrar la mejor secuencia de posibles etiquetas $\mathbf{e} = (e_1, e_2, ...e_r)$, en donde cada $e_i \in$ { Verbo, Sustantivo, Adjetivo, Adverbio, Determinantes, Pronombre, Conjunción, Interjección, Preposición, Puntuación, Numeral, FechaHora }, que se encuentra definida por

$$argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{n}, \mathbf{p}, \mathbf{t}, \mathbf{l}, \mathbf{m}, \mathbf{d}, \mathbf{s})$$

En donde $g_i \in$ { Masculino, Femenino, 0 }, $n_i \in$ { Singular, Plural, 0 }, $p_i \in$ { Primera, Segunda, Tercera , 0 }, $t_i \in$ { Presente, Pasado, Futuro, Imperfecto, Condicional, 0 }, $l_i \in$ { 1, 0 }, hace referencia al tamaño de la palabra, es decir si la cantidad de caracteres de la palabra es superior a 3, toma el valor de 1, caso contrario toma el calor 0, $m_i \in$ { 1, 0 } indica si la palabra comienza con mayúscula tomando el valor 1, caso contrario toma el valor 0, $d_i \in$ { 1, 0 }, en donde toma el valor 1 si la palabra contiene dígitos, caso contrario toma el valor 0, $s_i \in$ { 1, 0 } donde el valor 1 indica la presencia de almenos un símbolo en la palabra.

En este sentido para la oración "La intensidad de la señal luminosa", para i=6, esto es o_i= "luminosa", se presenta la siguiente función característica:

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{luminosa y } e_{i-1} = \text{sustantivo y } e_i = \text{adjetivo y} \\ & g_i = \text{femenino y } n_i = \text{singular y } t_i = 0 \text{ y } p_i = 0 \text{ y} \\ & l_i = 1 \text{ y } m_i = 0 \text{ y } d_i = 0 \text{ y } s_i = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

Donde $\mathbf{x}$ está conformado por o_i, g_i, n_i, t_i, p_i, l_i, m_i, d_i, s_i, en esta función característica. El modelo CRF, toma el observable de la posición actual, y se activa, esto es toma como valor 1, cuando se cumpla que g_i=femenino, n_i= singular, t_i=0, p_i=0, l_i=1, m_i=0, d_i=0, s_i=0, e_i= adjetivo y e_{i-1}= sustentivo, caso contrario toma el valor 0.

Otra función característica, que se crea a partir de la misma sentencia, es como sigue a continuación:

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{luminosa y } e_{i-1} = \text{sustantivo y } e_i = \text{adjetivo y} \\ & g_{i-1} = \text{femenino y } n_{i-1} = \text{singular y } t_{i-1} = 0 \text{ y } p_{i-1} = 0 \text{ y} \\ & l_{i-1} = 1 \text{ y } m_{i-1} = 0 \text{ y } d_{i-1} = 0 \text{ y } s_{i-1} = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

En la que se puede observar que se activa (toma el valor 1), teniendo en cuenta la etiqueta precedente, la etiqueta actual y las observaciones de la palabra $i-1$. Esto es posible dada la estructura del CRF.

3.3. Entrenamiento

Proceso encargado de realizar la tarea de aprendizaje del clasificador. Se utiliza para ello, la herramienta CRF++[1]. Recibe como entrada el conjunto preprocesado y una plantilla la cual contiene la configuración de dicho conjunto, es decir, especificando qué elementos son los observables y cuál es la correspondiente etiqueta. A partir de este entrenamiento se crea un archivo que contiene todas las funciones características con los respectivos pesos, los cuales fueron descritos en el capítulo 2. En este sentido se crean las funciones características que relacionan a cada etiqueta con la etiqueta inmediatamente anterior, y aquellas que relacionan a una etiqueta con los elementos observables. El archivo de salida es denominado el modelo producto del entrenamiento.

El archivo que corresponde a la plantilla, se estructura de la siguiente forma: dependiendo del modelo t que se esté trabajando, se formatea por columnas. Esto es, la primera columna tendrá la sentencia de entrada u oración, tokenizada en palabras,

[1] https://taku910.github.io/crfpp/

Cuadro 3.2: Plantilla para CRF++. Tomado de AnCora.

Palabra	Género	Etiqueta POS Tag
La	f	d
intensidad	f	n
de	0	s
la	f	d
señal	f	n
luminosa	f	a
es	0	v
idéntica	f	a
en	0	s
cada	c	d
uno	m	p
de	0	s
los	m	d
recuadros	m	n

es decir una palabra por cada fila. Las siguientes columnas contendrán (dependiendo del modelo t) las características de cada palabra. La última columna tendrá la etiqueta POS tag correspondiente a esa palabra, tal y como se visualiza en la tabla 3.2, para el modelo que contiene la palabra y la característica género.

Teniendo en cuenta la tabla 3.2, los Unigramas que se crean teniendo en cuenta la palabra actual, entonces, en la oración 'La intensidad de la señal luminosa', donde la palabra actual es 'luminosa', tienen la siguiente estructura:

func1 = if (output = a and feature='U01:f') return 1 else return 0

Es decir, que la función característica llamada 'func1' se activa con valor 1, cuando el género de la palabra actual sea 'f', que significa 'femenino' y la etiqueta POS Tag sea 'a', que significa 'adjetivo'.

La creación de los Bigramas se realiza mediante la especificación de la letra 'B' en la plantilla, con el objetivo que se generen todas las combinaciones de la etiqueta POS Tag actual y la etiqueta POS Tag anterior.

3.4. Clasificador

Como se observa en la figura 3.1, este proceso es el encargado de recibir como entrada dos archivos, el primero contiene la sentencia preprocesada, en donde cada una de las palabras va acompañada por sus características morfológicas; el segundo archivo contiene el modelo resultado del entrenamiento, el cual contiene todas las funciones características, con los correspondientes parámetros. Con estos dos elementos, se encuentra la distribución de etiquetas lingüísticas, mas probable, para esa sentencia. Se utiliza para el mismo, la herramienta CRF++[2]., la cual soporta el modelo CRF.

[2]https://taku910.github.io/crfpp/

4. Resultados

Teniendo en cuenta el proceso investigativo, del cual se habló en el primer capítulo, en esta sección se describen los modelos propuestos, así como los experimentos a los cuales se sometieron, bajo CRF para el español. Se presentan los resultados de la ejecución de ellos, y se los analiza confrontándolos con los resultados obtenidos de la ejecución del modelo HMM.

4.1. Diseño de Experimentos

Uno de los objetivos del proyecto es analizar el rendimiento del modelo CRF para el idioma español, y contrastarlo con el rendimiento, proporcionado por el modelo HMM. Para logar este objetivo, y teniendo en cuenta las características seleccionadas, se diseñó un conjunto de experimentos, los cuales se detallan a continuación.

En esta dirección se utilizó AnCora, el cual es un corpus que posee aproximadamente 500000 palabras, distribuidas en 10000 frases. Dicho conjunto contiene sentencias utilizadas para el entrenamiento, por lo que para el testeo se uso validación cruzada [31], mediante el cual se divide el conjunto en k subconjuntos. El proceso se realiza duratnte k iteraciones, en cada iteración se toma un subconjunto k_i, distinto de los tomados en las anteriores iteraciones. Ese conjunto es utilizado para el testeo. Con los restantes $k-1$ subconjuntos se forma un solo conjunto, el cual es utilizado para el entrenamiento.

Para la definición de los elementos observables, del modelo CRF, se utilizó escalado de características, el cual se soportó en [30], proceso mediante el cual se experimenta con distintas combinaciones de características de las palabras. Se experimentó con un total de 18 combinaciones o modelos, cada uno de los cuales contiene una configuración distinta de características. Como se detallan a continuación:

1. Modelo base. El modelo base cuenta con el token (la palabra), y ninguna característica morfológica de la misma, como se observa en la figura 4.1. Formalmente, se cuenta con un conjunto de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, se desea maximizar la probabilidad de la secuencia de etiquetas $\mathbf{e} = (e_1, e_2, ...e_r)$ dadas las observaciones, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o})$. El conjunto de entrenamiento contiene, por consiguiente, las palabras y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene únicamente las palabras. A partir de este modelo se irán incluyendo al mismo, las características morfológicas de las cuales se habló en el capítulo 2 y las características locales a la palabra, creando de esta forma nuevos modelos. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, en este sentido, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa", la palabra actual sea "intensidad", la etiqueta anterior sea "determinante", y la etiqueta actual sea "sustantivo", entonces la función característica se activa con valor 1. El modelo CRF activa esa función, cada vez que se encuentra esa observación, dentro del contexto de la secuencialidad.

   Una función característica creada a partir de este modelo es como se visualiza a continuación:

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

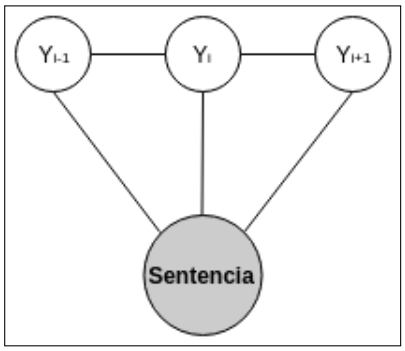


Figura 4.1: Modelo base

2. MB + una característica. Incluye al modelo base y una característica morfológica. Al modelo base, se le agrega la característica género, como se observa en la figura 4.2(a). Es decir a la secuencia $\mathbf{o} = (o_1, o_2, ...o_r)$ se le agrega la secuencia $\mathbf{g} = (g_1, g_2, ...g_r)$, en donde se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ dadas las observaciones y los géneros, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g})$. El conjunto de entrenamiento está conformado por las palabras, el género de cada una de ellas y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras y sus correspondiente géneros. Las funciones características se crearon a partir de plantillas definidas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por tanto, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa", la palabra actual sea "intensidad", la etiqueta sea "sustantivo", la etiqueta anterior sea "determinante", y el género sea femenino, entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo y } g_i = \text{femenino} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

Igualmente se le anexa al modelo base, la característica número que corresponde a la palabra, tal y como se observa en la figura 4.2(b). Esto es, a la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$ con la secuencia de etiquetas $\mathbf{n} = (n_1, n_2, ...n_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones y los números de las palabras, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{n})$. En este orden de ideas, el conjunto de entrenamiento está conformado por las palabras, el número de cada palabra y sus correspondientes etiquetas POS Tag. El conjunto de testeo conformado por las palabras y el número de cada una de ellas. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por tanto, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa", la palabra actual sea "intensidad", la etiqueta sea "sustantivo", la etiqueta que le precede sea "determinante", y el número sea singular, entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

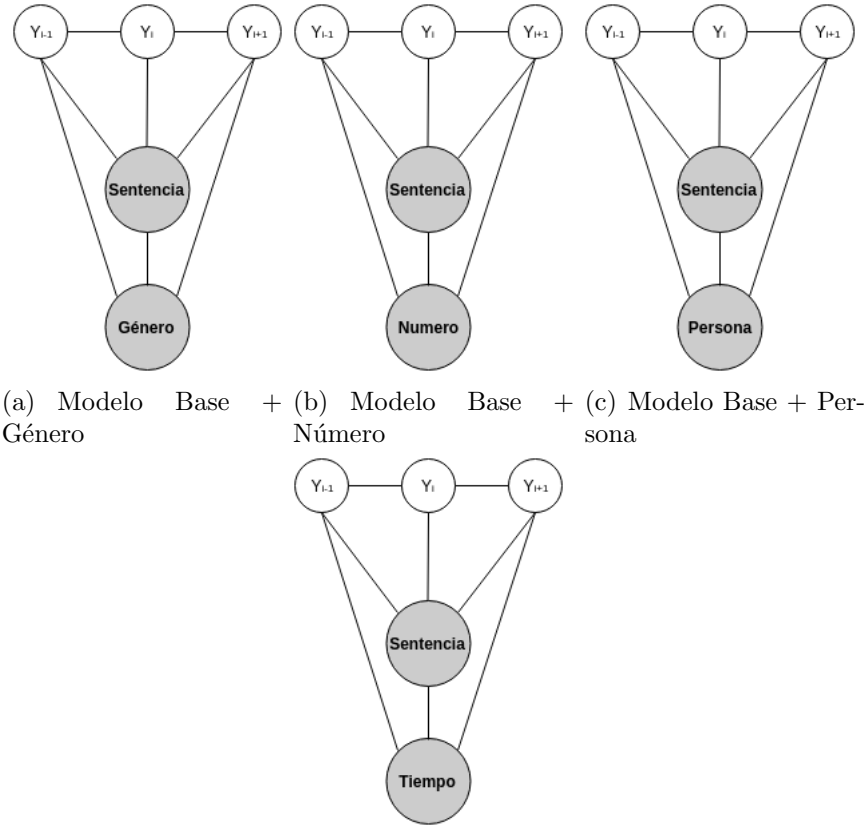
$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo y } n_i = \text{singular} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

Otro modelo creado a partir del modelo base, es el que se observa en la figura 4.2(c), el cual está conformado por la palabra y la característica morfológica persona de la misma. En otras palabras, se cuenta con la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$ y con la secuencia de etiquetas $\mathbf{p} = (p_1, p_2, ...p_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones y la característica morfológica persona, de cada palabra, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{p})$. En esta dirección, el conjunto de entrenamiento contiene las palabras, la característica persona, de cada palabra y sus correspondientes

etiquetas POS Tag. El conjunto de testeo conformado por las palabras y sus correspondientes características persona. Las funciones características se crearon a partir de plantillas definidas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por tanto, cuando se presente la secuencia de observaciones "la pantalla se queda", la palabra actual sea "se", la etiqueta sea "pronombre", la característica persona sea sea "tercera", y la etiqueta que le precede sea "sustantivo", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{se y } e_{i-1} = \text{sustantivo y } e_i = \text{pronombre y} \\ & p_i = \text{tercera} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

El cuarto modelo que surge, a partir del modelo base, es el que se presenta en la figura 4.2(d), el cual contiene la palabra y su correspondiente tiempo. En donde, teniendo la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$ y la secuencia $\mathbf{t} = (t_1, t_2, ...t_r)$ se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ dadas las observaciones y el tiempo de cada observación, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{t})$. El conjunto de entrenamiento conformado por las palabras, la característica tiempo, de cada palabra y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras y el tiempo de cada una de ellas. Las funciones características se defineron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por tanto, cuando se presente la secuencia de observaciones "la pantalla se queda", la palabra actual sea "queda", la etiqueta sea "verbo", el tiempo sea "presente", y la etiqueta que le precede sea "pronombre", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.



(a) Modelo Base + Género

(b) Modelo Base + Número

(c) Modelo Base + Persona

(d) Modelo Base + Tiempo

Figura 4.2: Modelo Base + una característica

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{queda y } e_{i-1} = \text{pronombre y } e_i = \text{verbo y} \\ & t_i = \text{presente} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

3. MB + dos características. Este modelo está compuesto por el modelo base y dos características morfológicas. A continuación se presentan todas las combinaciones presentes, con la palabra y sus correspondientes características morfológicas.

   En la figura 4.3(a) se observa la palabra con el género de la misma y el núme-

ro de ésta. Esto es, a la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{g} = (g_1, g_2, ...g_r)$ y con la secuencia de etiquetas $\mathbf{n} = (n_1, n_2, ...n_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, los géneros y los números de las palabras, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{n})$. El conjunto de entrenamiento está conformado por las palabras, el género de cada una de ellas al igual que su número y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, los géneros y los números de cada una de ellas. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por tanto, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa", la palabra actual sea "intensidad", la etiqueta sea "sustantivo", la etiqueta que la precede sea "determinante", el género sea "femenino", y el número sea "sigular", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo y } g_i = \text{femenino y } n_i = \text{singular} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

El modelo que incluye la palabra con su género y la correspondiente persona, se visualiza en la figura 4.3(b). Esto es, a la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{g} = (g_1, g_2, ...g_r)$ y la secuencia de etiquetas $\mathbf{p} = (p_1, p_2, ...p_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, el género y la característica morfológica persona de cada observación, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{p})$. El conjunto de entrenamiento está conformado por las palabras, el género de cada una de ellas al igual que su característica persona y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, los géneros y las características

persona, de cada una de ellas. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por tanto, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa", la palabra actual sea "intensidad", la etiqueta sea "sustantivo", la etiqueta que la precede sea "determinante", el género sea "femenino", y la persona sea "0", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo y } g_i = \text{femenino y } p_i = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

El modelo compuesto por la palabra, el género y el tiempo de la misma, se presenta en la figura 4.3(c). En otras palabras, se cuenta con la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{g} = (g_1, g_2, ...g_r)$ y con la secuencia de etiquetas $\mathbf{t} = (t_1, t_2, ...t_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, el género y el tiempo de cada palabra, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{t})$. En este sentido, el conjunto de entrenamiento contiene las palabras, el género de cada una de ellas al igual que su tiempo y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, los géneros y los tiempos de cada palabra. Las funciones características se crearon a partir de plantillas definidas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por tanto, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa", la palabra actual sea "intensidad", la etiqueta sea "sustantivo", la etiqueta que la precede sea "determinante", el género sea "femenino", y el tiempo sea "0", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo y } g_i = \text{femenino y } t_i = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

En la figura 4.3(d) se presenta el modelo base junto con el número y la persona de la palabra. En donde, teniendo la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, la secuencia $\mathbf{n} = (n_1, n_2, ...n_r)$ y la secuencia $\mathbf{p} = (p_1, p_2, ...p_r)$ se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ dadas las observaciones, el número y la persona de cada observación, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{n}, \mathbf{p})$. En este orden de ideas, el conjunto de entrenamiento está conformado por las palabras, el número de cada una de ellas al igual que su característica persona, de cada una de ellas y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, los números y las personas de cada palabra. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por tanto, cuando se presente la secuencia de observaciones "la pantalla se queda", la palabra actual sea "se", la etiqueta sea "pronombre", la etiqueta que la precede sea "sustantivo", la persona sea "tercera", y el número sea "0", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{se y } e_{i-1} = \text{sustantivo y} \\ & e_i = \text{pronombre y } p_i = \text{tercera y } n_i = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

El modelo conformado por la palabra, el número y el tiempo de la misma, se visualiza en la figura 4.3(e). Es decir, a la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{n} = (n_1, n_2, ...n_r)$ y con la secuencia

de etiquetas $\mathbf{t} = (t_1, t_2, ...t_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, los números y los tiempos de las palabras, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{n}, \mathbf{t})$. El conjunto de entrenamiento contiene las palabras, el número de cada una de ellas al igual que el tiempo y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, los números y los tiempos de cada una de ellas. Las funciones características se crearon a partir de plantillas definidas con la característica x_{i+j} donde $-2 \leq j \leq 2$, luego, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa", la palabra actual sea "intensidad", la etiqueta sea "sustantivo", la etiqueta que la precede sea "determinante", el número sea "singular", y el tiempo sea "0", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo y } n_i = \text{singular y } t_i = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

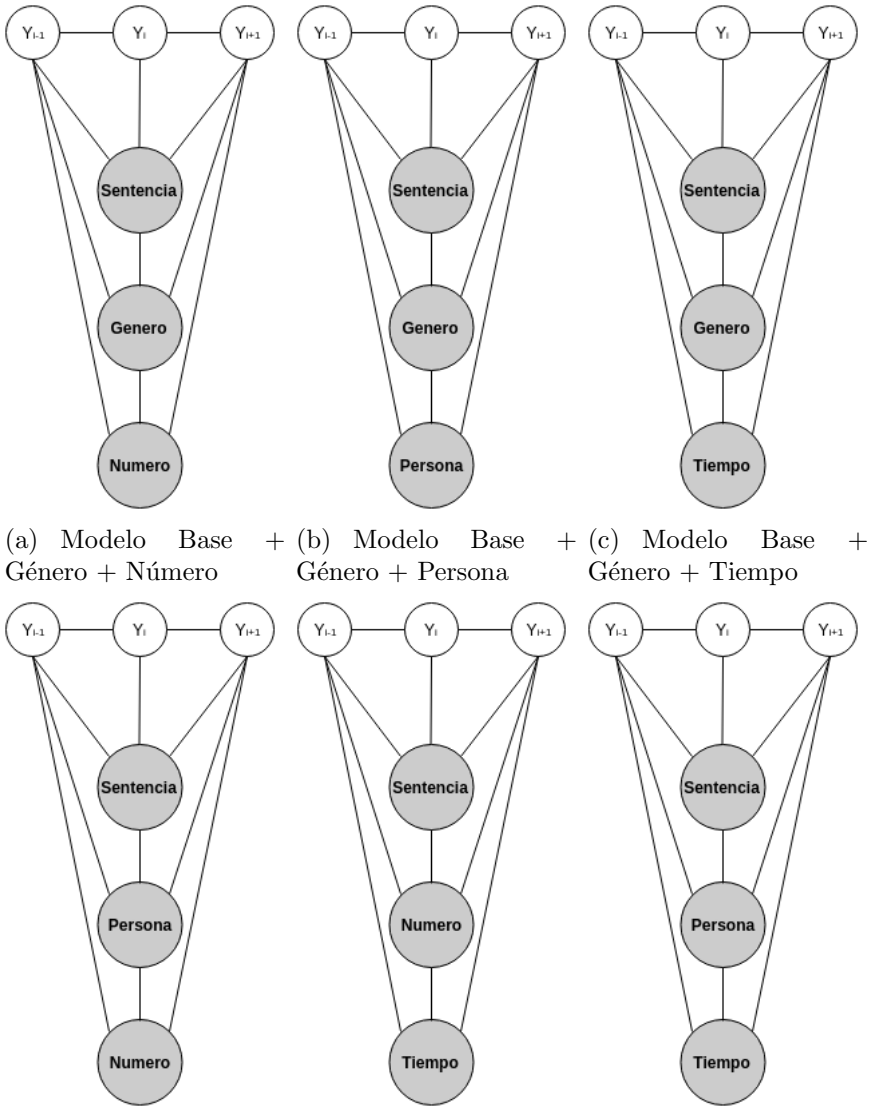
El último modelo, que se encuentra dentro de los que presentan 2 características adicionales, hace referencia al que, además del modelo base, incluye a la persona y el tiempo de la palabra, como se observa en la figura 4.3(f). En otras palabras, se cuenta con la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{p} = (p_1, p_2, ...p_r)$ y con la secuencia de etiquetas $\mathbf{t} = (t_1, t_2, ...t_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, la persona y el tiempo de cada palabra, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{p}, \mathbf{t})$. En esta dirección, el conjunto de entrenamiento está conformado por las palabras, la característica persona, de cada una de ellas al igual que su tiempo, y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, las personas y el tiempo de cada palabra. Las

funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, entonces, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa es", la palabra actual sea "es", la etiqueta sea "verbo", la etiqueta que la precede sea "adjetivo", el tiempo sea "presente", y la persona sea "tercera", la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{es y } e_{i-1} = \text{adjetivo y} \\ & e_i = \text{verbo y } t_i = \text{presente y } p_i = \text{tercera} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

4. MB + tres características. Conjunto de modelos conformados por el modelo base y tres caraterísticas adicionales al mismo.

   El primer modelo que se presenta es el conformado por la palabra, el género, el número y la persona de la misma, como se observa en la figura 4.4(a). Esto es, a la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{g} = (g_1, g_2, ...g_r)$, la secuencia de etiquetas $\mathbf{n} = (n_1, n_2, ...n_r)$ y la secuencia de etiquetas $\mathbf{p} = (p_1, p_2, ...p_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, el género, el número y la característica morfológica persona de cada observación, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{n}, \mathbf{p})$. En este orden de ideas, el conjunto de entrenamiento está conformado por las palabras, el género de cada una de ellas al igual que su número, la característica persona de cada palabra y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, los géneros, los números y la característica persona de cada una de ellas. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, luego, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa",



(a) Modelo Base + Género + Número

(b) Modelo Base + Género + Persona

(c) Modelo Base + Género + Tiempo

(d) Modelo Base + Número + Persona

(e) Modelo Base + Número + Tiempo

(f) Modelo Base + Persona + Tiempo

Figura 4.3: Modelo Base + dos característica

la palabra actual sea "intensidad", la etiqueta sea "sustantivo", la etiqueta que la precede sea "determinante", el género sea "femenino", el número sea "sigular", y la persona sea "0", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo y } g_i = \text{femenino y } n_i = \text{singular y } p_i = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

El siguiente modelo está conformado por el modelo base, el género, el número y el tiempo de la palabra, como se visualiza en la figura 4.4(b). Es decir, con la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{g} = (g_1, g_2, ...g_r)$, con la secuencia de etiquetas $\mathbf{n} = (n_1, n_2, ...n_r)$ y con la secuencia de etiquetas $\mathbf{t} = (t_1, t_2, ...t_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, el género, el número y el tiempo de cada observación, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{n}, \mathbf{t})$. El conjunto de entrenamiento contiene las palabras, el género de cada una de ellas, al igual que su número, su tiempo y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, los géneros, los números y los tiempos de cada una de ellas. Las funciones características se crearon a partir de plantillas definidas con la característica x_{i+j} donde $-2 \leq j \leq 2$, en este sentido, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa", la palabra actual sea "intensidad", la etiqueta sea "sustantivo", la etiqueta que la precede sea "determinante", el género sea "femenino", el número sea "sigular", y el tiempo sea "0", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{intensidad y } e_{i-1} = \text{determinante y} \\ & e_i = \text{sustantivo y } g_i = \text{femenino y } n_i = \text{singular y } t_i = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

En la figura 4.4(c) se presenta al modelo conformado por la palabra, el género, la persona y el tiempo de la misma. Formalmente, teniendo la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, la secuencia de etiquetas $\mathbf{g} = (g_1, g_2, ...g_r)$, con la secuencia de etiquetas $\mathbf{p} = (p_1, p_2, ...p_r)$ y la secuencia de etiquetas $\mathbf{t} = (t_1, t_2, ...t_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, el género, la persona y el tiempo de cada observación, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{p}, \mathbf{t})$. En esta dirección, el conjunto de entrenamiento está conformado por las palabras, el género, la característica persona, de cada una de ellas al igual que su tiempo, y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, el género, la característica persona y el tiempo de cada palabra. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, entonces, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa es", la palabra actual sea "es", la etiqueta sea "verbo", la etiqueta que la precede sea "adjetivo", el género sea "0", el tiempo sea "presente", y la persona sea "tercera", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

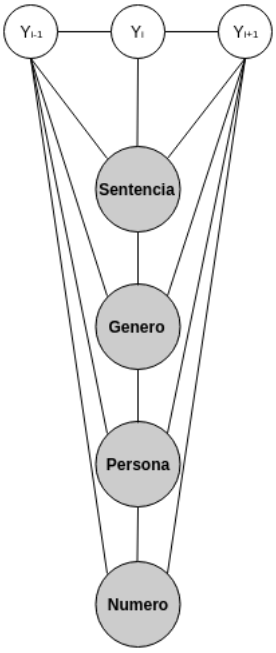
$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{es y } e_{i-1} = \text{adjetivo y } e_i = \text{verbo y} \\ & g_i = 0 \text{ y } t_i = \text{presente y } p_i = \text{tercera} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

En el último modelo de este grupo se encuentra al que está conformado por el modelo base, el número, la persona y el tiempo de la palabra, como se observa

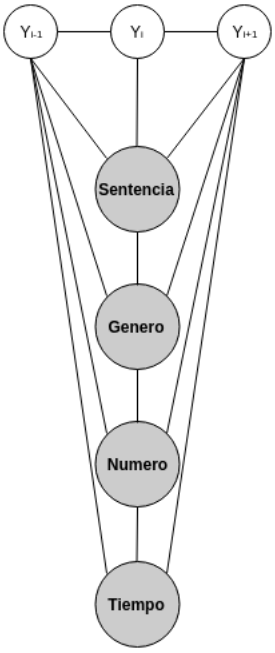
en la figura 4.4(d). En donde, teniendo la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{n} = (n_1, n_2, ...n_r)$, con la secuencia de etiquetas $\mathbf{p} = (p_1, p_2, ...p_r)$ y con la secuencia de etiquetas $\mathbf{t} = (t_1, t_2, ...t_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, el número, la persona y el tiempo de cada observación, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{n}, \mathbf{p}, \mathbf{t})$. Para este modelo el conjunto de entrenamiento contiene las palabras, su número, la característica persona de cada una de ellas, al igual que su tiempo y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, el número, la característica persona y el tiempo de cada palabra. Las funciones características se crearon a partir de plantillas definidas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por lo tanto, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa es", la palabra actual sea "es", la etiqueta sea "verbo", la etiqueta que la precede sea "adjetivo", el número sea "singular", el tiempo sea "presente", y la persona sea "tercera", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{es y } e_{i-1} = \text{adjetivo y } e_i = \text{verbo y} \\ & n_i = \text{singular y } t_i = \text{presente y } p_i = \text{tercera} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

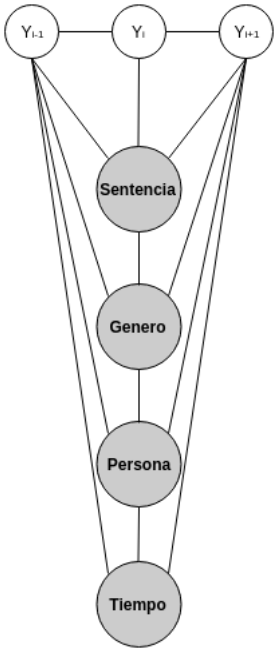
5. Modelo Base + cuatro características. El primer modelo que se encuentra aquí es el que está conformado por la palabra, el género, el número, la persona y el tiempo de la misma, como se visualiza en la figura 4.5(a). Esto es, a la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{g} = (g_1, g_2, ...g_r)$, la secuencia de etiquetas $\mathbf{n} = (n_1, n_2, ...n_r)$, la secuencia de etiquetas $\mathbf{p} = (p_1, p_2, ...p_r)$ y la secuencia de etiquetas $\mathbf{t} = (t_1, t_2, ...t_r)$, se



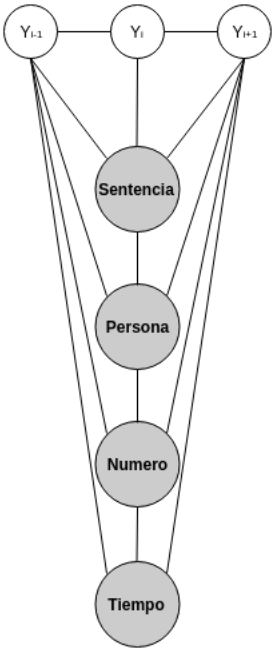
(a) Modelo Base + Género + Número + Persona



(b) Modelo Base + Género + Número + Tiempo



(c) Modelo Base + Género + Persona + Tiempo



(d) Modelo Base + Persona + Número + Tiempo

Figura 4.4: Modelo Base + tres características

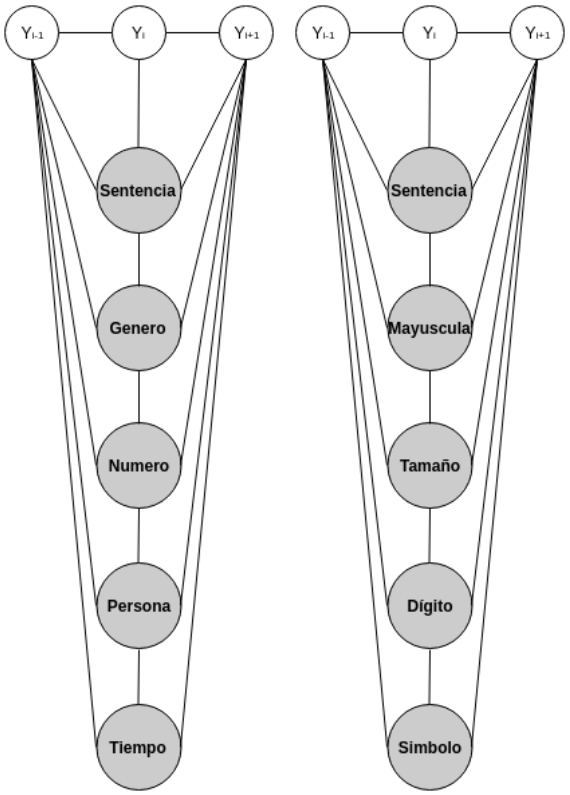
desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, el género, el número, la característica morfológica persona y el tiempo de cada observación, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{n}, \mathbf{p}, \mathbf{t})$. En este orden de ideas, el conjunto de entrenamiento contiene las palabras, sus géneros, sus números, la característica persona de cada una de ellas, al igual que su tiempo y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, sus géneros, sus números, la característica persona y el tiempo de cada palabra. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$, por lo tanto, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa es", la palabra actual sea "es", la etiqueta sea "verbo", la etiqueta que la precede sea "adjetivo", el género sea "0", el número sea "singular", el tiempo sea "presente", y la persona sea "tercera", entonces la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{es y } e_{i-1} = \text{adjetivo y } e_i = \text{verbo y} \\ & g_i = 0 \text{ y } n_i = \text{singular y } t_i = \text{presente y } p_i = \text{tercera} \\ 0 & \text{si no se cumple el caso} \end{cases}$$

El modelo que se presenta en la figura 4.5(b) contiene al modelo base e incluye características locales como, si la palabra comienza con mayúscula, el tamaño de la palabra, si la palabra contiene dígitos, si la plabara contiene símbolos. Es decir, con la secuencia de observaciones $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia de etiquetas $\mathbf{m} = (m_1, m_2, ...m_r)$, con la secuencia de etiquetas $\mathbf{l} = (l_1, l_2, ...l_r)$, con la secuencia de etiquetas $\mathbf{d} = (d_1, d_2, ...d_r)$ y con la secuencia $\mathbf{s} = (s_1, s_2, ...s_r)$, se desea maximizar $\mathbf{e} = (e_1, e_2, ...e_r)$ teniendo como entrada las observaciones, su correspondiente secuencia de tamaños, su correspondiente secuencia de tipos

de comienzo de palabra , su correspondiente secuencia de indicaciones de existencia de dígitos y su correspondiente secuencia de indicaciones de existencia de símbolos, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{m}, \mathbf{l}, \mathbf{d}, \mathbf{s})$. En esta dirección, el conjunto de entrenamiento contiene las palabras, sus tamaños, su secuencia de indicaciones de comienzo con mayúscula, su secuencia de indicaciones de existencia de dígito, su secuencia de indicaciones de existencia de símbolos y sus correspondientes etiquetas POS Tag. El conjunto de testeo contiene las palabras, sus tamaños, su secuencia de indicaciones de comienzo con mayúscula, su secuencia de indicaciones de existencia de dígito y su secuencia de indicaciones de existencia de símbolos. Las funciones características se definieron a partir de plantillas creadas con la característica x_{i+j} donde $-2 \leq j \leq 2$.

6. Modelo Completo. Modelo que contiene al modelo base, el género, el número, la persona y el tiempo de la palabra, además incluye características como, si la palabra comienza con mayúscula, el tamaño de la palabra, si la palabra contiene dígitos, si la plabara contiene símbolos. Todos estos elementos conforman los observables en el modelo CRF, como se presenta en la figura 4.6. Esto es dada una secuencia $\mathbf{o} = (o_1, o_2, ...o_r)$, con la secuencia $\mathbf{g} = (g_1, g_2, ...g_r)$, con la secuencia $\mathbf{n} = (n_1, n_2, ...n_r)$, con la secuencia $\mathbf{p} = (p_1, p_2, ...p_r)$, con la secuencia $\mathbf{t} = (t_1, t_2, ...t_r)$, con la secuencia $\mathbf{l} = (l_1, l_2, ...l_r)$, con la secuencia $\mathbf{m} = (m_1, m_2, ...m_r)$, con la secuencia $\mathbf{d} = (d_1, d_2, ...d_r)$ y con la secuencia $\mathbf{s} = (s_1, s_2, ...s_r)$, el objetivo es encontrar la mejor secuencia de posibles etiquetas $\mathbf{e} = (e_1, e_2, ...e_r)$, de la siguiente forma $argmaxp(\mathbf{e}|\mathbf{o}, \mathbf{g}, \mathbf{n}, \mathbf{p}, \mathbf{t}, \mathbf{l}, \mathbf{m}, \mathbf{d}, \mathbf{s})$. El conjunto de entrenamiento contiene las palabras, sus géneros, sus números, la característica persona de cada una de ellas, al igual que su tiempo, el tamaño, el tipo de comienzo de la palabra y la existencia o no, de dígitos y símbolos y sus correspondientes etiquetas POS Tag. El conjunto de testeo, por tanto, contiene las palabras, sus géneros, sus números, la característica persona, el tiempo de



(a) Modelo Base + Género + Número + Persona + Tiempo

(b) Modelo Base + Mayúscula + Tamaño + Dígito + Símbolo

Figura 4.5: Modelo Base + cuatro características

cada palabra, al igual que el tamaño, el tipo de comienzo de la palabra y la existencia o no, de dígitos y símbolos. Las funciones características se crearon a partir de plantillas definidas con la característica x_{i+j} donde $-2 \leq j \leq 2$, entonces, cuando se presente la secuencia de observaciones "La intensidad de la señal luminosa es", la palabra actual sea "es", la etiqueta sea "verbo", la etiqueta que la precede sea "adjetivo", el género sea "0", el número sea "singular", el tiempo sea "presente", la persona sea "tercera", el tamaño sea "0", el tipo de comienzo de la palabra sea "0", la existencia de dígitos sea "0", y la existencia de símbolos sea "0", la función característica se activa, tomando el valor 1, dentro del contexto de la secuencialidad.

$$f(e_i, e_{i-1}, \mathbf{x}) = \begin{cases} 1 & \text{si } o_i = \text{es y } e_{i-1} = \text{adjetivo y } e_i = \text{verbo y} \\ & g_i = 0 \text{ y } n_i = \text{singular y } t_i = \text{presente y } p_i = \text{tercera} \\ & l_i = 0 \text{ y } m_i = 0 \text{ y } d_i = 0 \text{ y } s_i = 0 \\ 0 & \text{si no se cumple el caso} \end{cases}$$

Las variables que permiten evaluar el rendimiento del modelo, son Accuracy, Recall, Precision y Score FB1. El cálculo de las mismas se lo realiza teniendo en cuenta la tabla 4.1, donde el análisis se lo realiza por cada clase o categoría lingüística. TP corresponde con los casos cua clase era positiva y fuéron clasificados correctamente. TN son los casos que eran negativos y no fueron clasificados de forma positiva. FN son los casos que eran positivos y fueron etiquetados de forma negativa. FP, casos que eran negativos y fueron clasificados de forma positiva.

4.2. Análisis y Discusión de Resultados

A continuación se presentan los resultados de la ejecución de cada uno de los modelos planteados con CRFs, además de la ejecución del modelo base con HMMs.

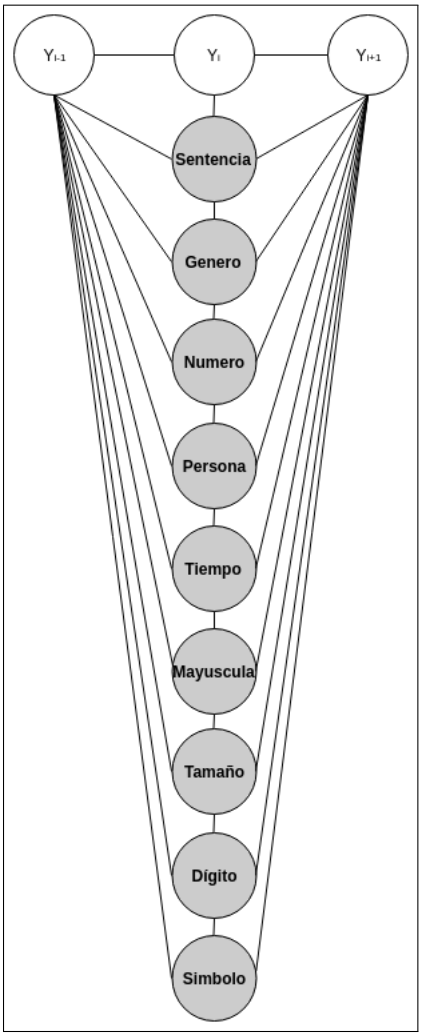


Figura 4.6: Modelo Completo

Cuadro 4.1: Variables tenidas en cuenta para medir el rendimiento de cada modelo

Variable	Descripción	Fórmula
Precision	Porción de buena predicción para los positivos	TP/(TP + FP)
Recall	Porción de casos positivos que fueron identificados correctamente	TP/(TP + FN)
Accuracy	Cantidad de casos positivos y negativos, que fueron predecidos correctamente, con respecto a todos los casos clasificados.	(TP + TN) / (TP + TN +FP+FN)
Score FB1	Media armónica entre la precisión y la recuperación.	2 * (P * R) /(P + R)

Tanto para la primera parte, como para la segunda se calculó el valor de cada una de las variables definidas para la presente investigación, esto es accuracy, precision, recall y FB1. Igualmente se presentan los resultados del tiempo de entrenamiento para cada modelo, bajo CRF.

En la tabla 4.2 se puede observar los resultados de evaluar cada uno de los modelos propuestos, utilizando CRF. Para cada uno de ellos se calculó el Accuracy, la Precision, el Recall y el Score FB1. El cálculo de las mencionadas variables es como se muestra en la tabla 4.1.

El mejor rendimiento, expresado en términos de accuracy, precision, recall y score FB1, lo presenta el modelo completo. Con el modelo completo se obtiene el mejor rendimiento que los demás, profundizando en este modelo y como se observa en la tabla 4.5, se concluye que los adjetivos son los que tienen la menor precisión y recall. Es decir, los que mas errores en la clasificación tienen y por ende, influyen en que el modelo no tenga valores de precisión, accuraccy, score FB1 y recall mas altos. Analizando los resultados de clasificación para los adjetivos, se observa que 82.9 % de los adjetivos son clasificados de forma correcta, mientras que el 17.1 % se clasificaron de forma incorrecta. Dentro de la categoría de los adjetivos, se encontró como palabras nuevas el 33.1 %, de los cuales se clasificó correctamente el 72.4 %, e incorrectamente

Cuadro 4.2: Resultados de cada modelo

Modelo - Variable	Accuracy %	Precision %	Recall %	FB1 %
Modelo Base	92.56	91.33	91.60	91.47
Modelo base + género	94.95	94.01	94.09	94.05
Modelo base + número	93.93	92.92	93.15	93.04
Modelo base + persona	93.87	92.73	93.06	92.89
Modelo base + tiempo	93.81	92.68	92.97	92.83
Modelo base + género + número	95.67	94.88	95.01	94.95
Modelo base + género + persona	95.66	94.86	95.01	94.93
Modelo base + género + tiempo	95.55	94.75	94.89	94.82
Modelo base + número + persona	95.29	94.44	94.59	94.51
Modelo base + número + tiempo	95.29	94.45	94.61	94.53
Modelo base + tiempo + persona	93.91	92.80	93.10	92.95
Modelo base + género + número + persona	95.71	94.90	95.05	94.97
Modelo base + género + número + tiempo	95.65	94.85	94.98	94.91
Modelo base + género + persona + tiempo	95.62	94.83	94.97	94.90
Modelo base + número + persona + tiempo	95.37	94.54	94.69	94.62
Modelo base + género + número + persona + tiempo	95.69	94.89	95.03	94.96
Modelo base + mayúscula + tamaño + dígito + símbolo	92.56	91.33	91.60	91.47
Modelo completo	96.39	95.73	95.89	95.81

Cuadro 4.3: Resultados modelo HMMs

Modelo - Variable	Accuracy %	Precision %	Recall %	FB1 %
HMMs	96.23	95.41	95.46	95.44

Cuadro 4.4: Tiempo de entrenamiento de cada modelo

Modelo	Tiempo entrenamiento (segundos)
Modelo Base	600.69
Modelo base + género	1066.21
Modelo base + número	1113.82
Modelo base + persona	1379.20
Modelo base + tiempo	1408.74
Modelo base + género + número	1282.43
Modelo base + género + persona	1433.61
Modelo base + género + tiempo	1510.93
Modelo base + número + persona	1491.88
Modelo base + número + tiempo	1507.24
Modelo base + tiempo + persona	1969.82
Modelo base + género + número + persona	1671.82
Modelo base + género + número + tiempo	1680.66
Modelo base + género + persona + tiempo	1857.89
Modelo base + número + persona + tiempo	1857.61
Modelo base + género + número + persona + tiempo	2034.69
Modelo base + mayúscula + tamaño + dígito + símbolo	593.06
Modelo completo	2612.74

Cuadro 4.5: Resultados por categoría para el modelo completo

Categoría	Precisión	Recall	FB1
Adjetivos	84.83	82.33	96.18
Conjunciones	97.99	97.42	97.70
Determinantes	98.27	98.63	98.45
Puntuación	99.78	99.87	99.83
Sustantivos	92.85	95.32	94.06
Pronombres	98.55	94.88	96.67
Adverbios	92.28	92.07	92.17
Preposiciones	98.99	98.89	98.94
Verbos	95.83	95.69	95.76
Numerales	96.67	89.19	92.07

el 27.6 %. Dentro de los adjetivos clasificados de forma incorrecta se encuentra que la mayoría, esto es el 91.6 %, son clasificados como sustantivos.

En general para el modelo completo, se clasifica correctamente el 96.5 %, e incorrectamente el 3.5 %. Para este modelo se cuenta con un total de 15.6 % de palabras nuevas, siendo clasificadas correctamente el 90.6 % de ellas e incorrectamente el 9.3 %. Las demás categorías lingüísticas tienen valores superiores al 92.28 % en precisión, 89 % en recall y 92.07 % en SCORE FB1.

Análisis del Accuracy

Como se observa en la figura 4.7(a), y en la tabla 4.2, el porcentaje mínimo fue de 92.56 % correspondiente al modelo base, así como también para el modelo17. El porcentaje máximo fué de 96.39 %, correspondiente al modelo completo. Al calcular esta variable con el modelo base se obtuvo un porcentaje que inmediatamente subió, cuando se agregó una característica al modelo. Como se observa en el caso del tiempo, el número, la persona y el género, obteniendo un mayor valor con la característica género. El porcentaje aumenta cuando se trabajan con 2, ó con 3 características.

Igualmente se mantiene cuando se trabajan con las 4 características: género, número,persona y tiempo. Al trabajar con el modelo 17, el porcentaje baja notablemente. El mejor porcentaje se obtiene con el modelo completo, el cual incluye todas las características. Como se pudo observar en la sección del diseño de los experimentos, cada vez que se aumenta una característica a un modelo, las funciones características correspondientes al mismo se vuelven mas complejas, es decir para que se active una de ellas, se tienen que cumplir muchas condiciones que serían evaluadas en el proceso del testeo. Sin embargo, el archivo de testeo, a medida que aumentan las características, contiene mucha mas información que el mismo archivo para el modelo base. El alto porcentaje en el valor del accuracy en el modelo completo indica que la cantidad de casos clasificados correctamente, tanto positivos como negativos, para cada una de las clases, con respecto al total de casos clasificados es superior que el presentado por el modelo base.

Análisis de la Presición

Como se visualiza en la figura 4.7(b), y en la tabla 4.2, el porcentaje mínimo calculado, es de 91.33 %, correspondiente a los modelos, base y 17. El valor máximo obtenido es de 95.72 %, correspondiente al modelo completo. Del modelo base, al modelo base incluyendo una característica, se observa un incremento en el porcentaje. Pero este se mantiene con valores similares, con los siguientes modelos. El modelo completo, tiene el mayor porcentaje, ya que incluye todas las caracerísticas de cada palabra. Al igual que el análisis realizado con el accuracy, cada vez que se aumenta una característica, el número de condiciones presente en las funciones características es superior, haciendo que la evaluación de estas en el testeo sea mas compleja. Pero hay que tener en cuenta que el conjunto de testeo contiene mas características que el archivo del modelo base, cada vez que se aumenta una característica, lo que permite que se clasifiquen de una forma correcta, mayor número de casos positivos predecidos, con respecto al total de los casos predecidos como positivos.

Análisis del Recall

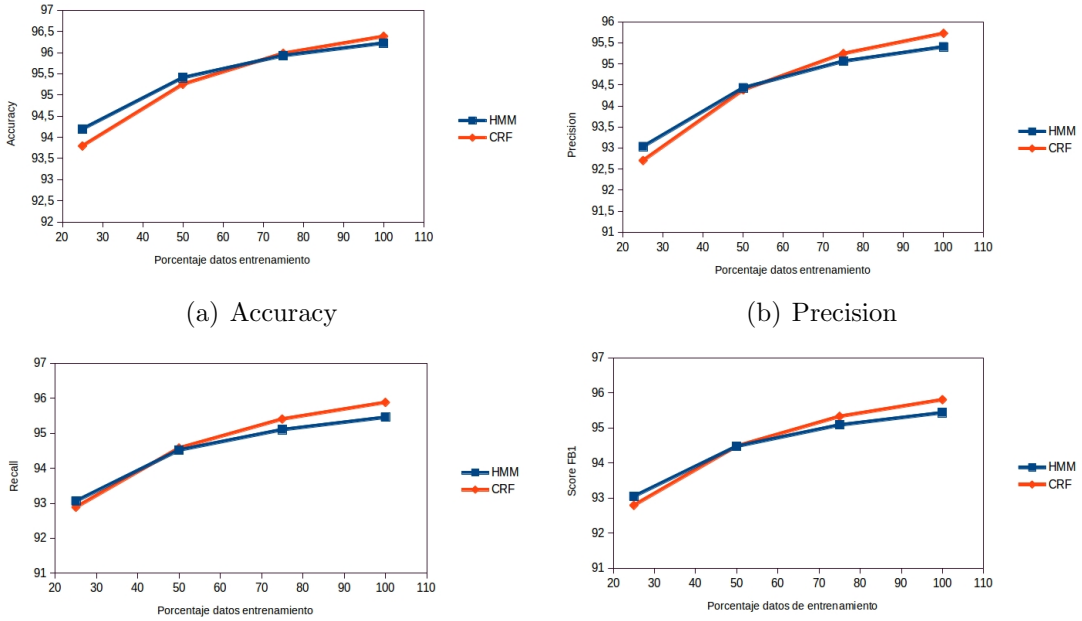
Como se observa en la figura 4.7(c), y en la tabla 4.2, el porcentaje mínimo corresponde al modelo base, con un valor de 91.60 %, y al modelo 17 con el mismo porcentaje. El modelo que marca un mayor valor porcentual, para esta variable, es el modelo completo, con un 95.88 %. Del modelo base, a los modelos que incluyen 2, 3 y 4 características, se observa un aumento porcentual. El análisis es similar que para las anteriores variables. La complejidad de las funciones características aumenta, en la medida que aumenta el número de características de las observaciones. En el proceso de testeo se evalúan dichas funciones características y para que sean activadas, tienen que cumplirse, un mayor número de condiciones, como es el caso del modelo completo. Esto permite que, para este modelo, el cual tiene el mayor número de características, su evaluación permita que se clasifiquen de forma correcta, un mayor número de casos positivos, sobre el total de los casos positivos presentes en el conjunto de testeo.

Análisis del Score FB1

Como se visualiza en la figura 4.7(d), y en la tabla 4.2, el menor porcentaje corresponde al modelo base, con un valor porcentual de 91.40 %, igualmente para el modelo17. El mejor valor porcentual, se obtuvo en el modelo completo, con un valor de 95.80 % . El modelo completo, al abarcar todas las caracterícticas, presenta un porcentaje mayor que los demás modelos. El análisis realizado para las variables de la precisión y del recall, se extiende para el score FB1, en que esta última presenta una media armónica de las dos variables, antes citadas. A medida que aumenta el número de características, como es el caso del modelo completo, la clasificación va a ser superior, dado que existe mas información, lo cual permite que se active un mayor número de funciones características, de las definidas para este modelo.

En la tabla 4.3, se puede observar los resultados de la ejecución del experimento bajo HMMs. En el mencionado modelo, se obtiene los siguientes resultados. Accuracy: 96.23 %, Presición: 95.41 %, Recall: 95.46 %, y Score FB1: 95.44. Este modelo

no tiene en cuenta las características de cada palabra, sin embargo alcanza un porcentaje bastante elevado.



(a) Accuracy (b) Precision

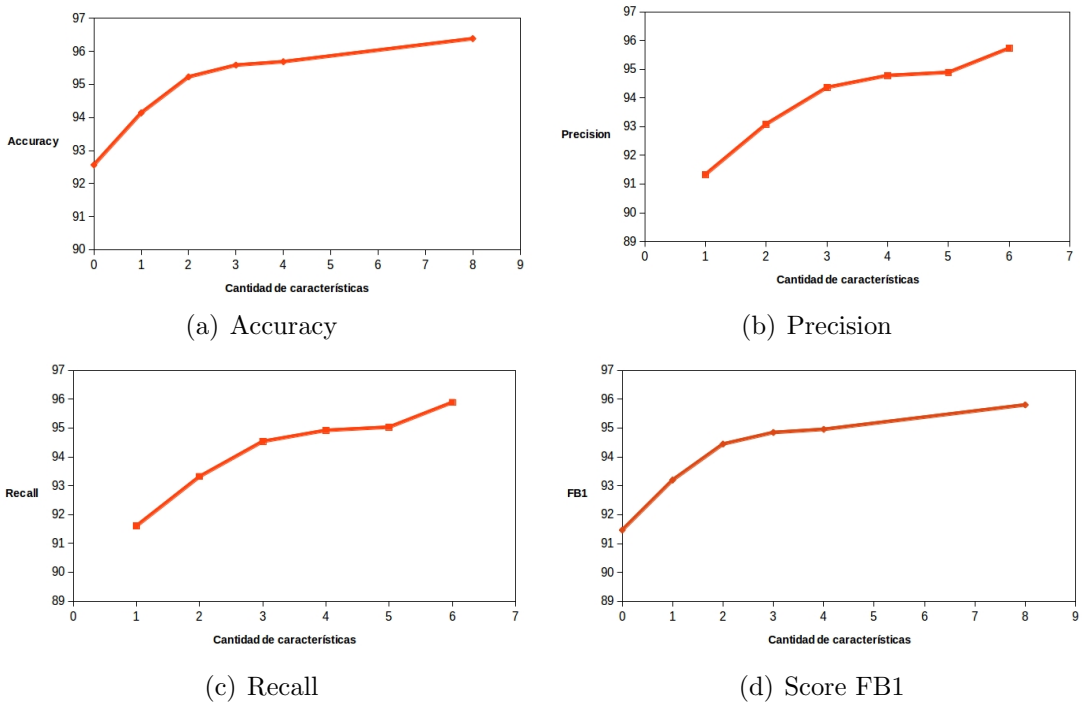
(c) Recall (d) Score FB1

Figura 4.7: CRF y HMM vs tamaño datos de entrenamiento

Tiempo de entrenamiento

El modelo base tarda 600.69 segundos en ser entrenado, siendo el que menos se demora. A medida que se va aumentando el número de características que se le agregan al modelo base, el tiempo de entrenamiento va subiendo, hasta obtener un tiempo de entrenamiento 2612.74 segundos, para el modelo completo. La información correspondiente a los tiempos de entrenamiento de cada modelo, se encuentra en la tabla 4.4.

Análisis del modelo CRF vs Cantidad de características En la figura 4.8(a), se visualiza que a medida que aumenta el número de características, dentro del modelo CRF, aumenta el accuracy, en la tarea del POS Tagging. Igualmente para la precision, como se visualiza en la figura 4.8(b), el porcentaje de la misma



(a) Accuracy (b) Precision

(c) Recall (d) Score FB1

Figura 4.8: Accuracy, Precision, Recall, FB1 vs Cantidad de características

se incrementa, a medida que se aumenta el número de características. También al incrementar el número de características, el porcentaje del recall sube, como se observa en la figura 4.8(c). Para el score FB1, a medida que aumenta el número de características, el porcentaje del mismo es mayor.

Análisis del modelo CRF vs el tamaño del corpus de entrenamiento

En cuanto al tamaño del corpus de entrenamiento de los modelos CRF y HMM se realiza el siguiente análisis. En la figura 4.7(a) se observa que al entrenar el modelo con un 25 % del tamaño total del corpus, se alcanza un accuracy inferior en el modelo CRF que en el modelo HMM. Sin embargo, cuando el modelo es entrenado con porcentajes superiores al 75 % del tamaño total del corpus, la accuracy es superior en CRF que en HMM. En la figura 4.7(b) se visualiza que cuando se entrena el modelo con un 25 % del tamaño total del corpus, se alcanza una precisión inferior en el modelo CRF que en el modelo HMM. No obstante, cuando se entrena el modelo con

porcentajes superiores al 75 % del tamaño total del corpus, la presición es superior en CRF que en HMM. En la figura 4.7(c) se observa claramente que al entrenar el modelo con porcentajes iguales o superiores al 50 % del tamaño total del corpus, el recall es superior en CRF que en HMM. Finalmente, para el score FB1 el análisis es similar que para el recall, teniendo como base para éste, la figura 4.7(d).

Teniendo en cuenta todas las variables, cuando se entrena con el total del corpus de entrenamiento, el modelo CRF es superior al modelo HMM en la tarea del POS Tagging. Se observa también que cuando el modelo es entrenado con el 25 % del total del corpus de entrenamiento, el modelo HMM alcanza porcentajes superiores, en las mencionadas variables, en la tarea del POS Tagging, esto se debe a que el modelo CRF tiene en cuenta todas las características de las palabras, además que tiene en cuenta toda la sentencia de entreada, por lo cual, entre mas palabras y sus características se disponga en el corpus de entrenamiento, mayor porcentaje en el etiquetado se alcanzará.

Teniendo en cuenta el anterior análisis, se comprueba la hipótesis planteada, bajo la cual se dijo, en un principio, que los porcentajes de etiquetado en la tarea del POS Tagging, utilizando el modelo CRF, son superiores a los porcentajes utilizados por el modelo HMM, en el idioma español, bajo el corpus de AnCora.

5. Conclusiones y trabajos futuros

Se presentó en este documento, el informe final de investigación, en donde se estudia el comportamiento en la tarea del POS Tagging, realizada por el Modelo CRF, y se lo contrastó con el modelo HMM en la misma tarea, para el idioma español, bajo el corpus AnCora.

A medida que se aumenta el número de características, en el modelo CRF, los niveles de accuracy, presición, recall y FB1 aumentan, en la tarea del POS Tagging, esto es debido a que el modelo CRF, para esta tarea, se soporta en toda la sentencia de entrada y en las características de los observables.

El proceso mas costoso en el modelo CRF hace referencia al tiempo de entrenamiento. A medida que aumenta el número de características, aumenta el tiempo de entrenamiento.

El modelo CRF alcanza mayores niveles de accuracy, presición, recall y FB1, que el modelo HMM, en la tarea de POS Tagging para el idioma español, utilizando el corpus AnCora.

Luego de haber alcanzado unos niveles de precisión, accuracy, recall y fb1, en el modelo CRF, superiores a los del modelo HMM, se recomienda realizar un estudio similar con otros modelos que resuelven el problema del secuence labelling, como lo son máquinas de soporte vectorial, modelo basado en máxima entropía, modelo basado en reglas, etc. Bajo el corpus AnCora, para el idioma español.

Bibliografía

[1] Friederike muller and birgit waibel. corpus linguistics - an introduction. https://portal.uni-freiburg.de/angl/seminar/abteilungen/sprachwissenschaft/ls_mair/corpus-linguistics/corpus-linguistics-an-introduction#basics.

[2] Milka villayandre llamazares. Lingüística Computacional II. Curso monográfico sobre Lingüística de corpus - Universidad de León.

[3] Neetu Aggarwal and Amandeep kaur Randhawa. A survey on part of speech tagging for indian languages. *International Journal of Computer Applications International Conference on Advancements in Engineering and Technology*, 2015.

[4] Graves Alex. *Supervised Sequence Labelling with Recurrent Neural Networks*. 2012.

[5] McCallum Andrew and Sutton Charles. An introduction on conditional random fields. *Foundations and Trends in Machine Learning: Vol. 4*, 2012.

[6] Sergio-Nabil Khayyat Arranz. Campos aleatorios condicionales. Master's thesis, Universidad Autónoma de Madrid Escuela Politécnica Superior Departamento de Ingeniería Informática, 2011.

[7] Ekbal Asif, Haque Rejwanul, and Bandyopadhyay Sivaji. Bengali part of speech

tagging using conditional random field. *Proceedings of Seventh International Symposium on Natural Language Processing*, 2007.

[8] Rakesh Balabantaray and Deepak Sahoo. An experiment to create parallel corpora for odia. *International Journal of Computer Applications Volume 67*, 2013.

[9] Anup Kumar Barman, Jumi Sarmah, and Shikhar Kr. Sarma. Pos tagging of assamese language and performance analysis of crf++ and fntbl approaches. *UKSIM '13 Proceedings of the 2013 UKSim 15th International Conference on Computer Modelling and Simulation*, 2013.

[10] Christopher Bishop. *Pattern Recognition and Machine Learning Sample_chapter*, chapter Graphical Models, pages 359 – 418. Springer, 2006.

[11] Thorsten Brants. Tnt - a statistical part-of-speech tagger. *ANLC 00 Proceedings of the sixth conference on Applied natural language processing*, 2000.

[12] Eric Brill. A simple rule-based part of speech tagger. *ANLC 92 Proceedings of the third conference on Applied natural language*, 1992.

[13] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Association for Computational Linguistics*, 1995.

[14] Parra Escartín Carla and Martínez Alonso Héctor. Choosing a spanish part-of-speech tagger for a lexically sensitive task. *Procesamiento del Lenguaje Natural Revista Numero 54*, 2015.

[15] Bishwa Ranjan Das, Smrutirekha Sahoo, Chandra Sekhar Panda, and Srikanta Patnaik. Part of speech tagging in odia using support vector machine. *Procedia Computer Science Volume 48*, 2015.

[16] Farwell David, Helmreich, Stephen, and Casper Mark. Spost a spanish part of speech tagger. *Procesamiento del lenguaje natural numero 17*, 1995.

[17] Giesbrecht Eugenie and Evert Stefan. Is part-of-speech tagging a solved task? an evaluation of pos taggers for the german web as corpus. *Fifth Web as Corpus workshop*, 2009.

[18] Zamora Francisco, Castro María José, España Salvador, and Tortajada Salvador. Adding morphological information to a connectionist part of speech tagger. *Current Topics in Artificial Intelligence*, 2009.

[19] Dietterich Thomas G. *Structural, Syntactic, and Statistical Pattern Recognition*, chapter Machine Learning for Sequential Data: A Review, pages 15–30. Springer Berlin Heidelberg, 2002.

[20] Alexander Gelbukh and Raul Morales Carrasco. Evaluation of tnt tagger for spanish. *Fourth Mexican International Conference on Computer Science*, 2003.

[21] Byrd R. H., Lu P., and Nocedal J. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 1995.

[22] Aldarmaki Hanan and Diab Mona. Robust part-of-speech tagging of arabic text. *The Second Workshop on Arabic Natural Language Processing*, 2015.

[23] Daniel Jurafsky and James Martin. *SPEECH and LANGUAGE PROCESSING An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall, 2009.

[24] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *18th*

International Conference on Machine Learning 2001 (ICML 2001), pages 282–289, 2001.

[25] Rabiner Lawrence and AT&T Bell Lab. Murray Hill. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE (Volume 77)*, 1989.

[26] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics - Special issue on using large corpora: II*, 1993.

[27] Andrew McCallum. Efficiently inducing features of conditional random fields. *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, 2003.

[28] Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. Maximum entropy markov models for information extraction and segmentation. *ICML 00 Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[29] Monachini Monica and Calzolari Nicoletta. *EAGLES Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and Corpora. A Common Proposal and Applications to European Languages.* Centre National de la Recherche Scientifique Paris, France, 1996.

[30] Refaeilzadeh Payam, Tang Lei, and Liu Huan. On comparison of feature selection algorithms. *AAAI Workshop - Vancouver, BC, Canada*, 2007.

[31] Refaeilzadeh Payam, Tang Lei, and Liu Huan Arizona State University. *Encyclopedia of Database Systems.* M. Tamer A zsu and Ling Liu. Springer., 2009.

[32] Awasthi Pranjal, Rao Delip, and Ravindran Balaraman. Part of speech tagging and chunking with hmm and crf. *NLP Association of India (NLPAI) Machine Learning Contest*, 2006.

[33] Avinesh PVS and Karthik G. Part of speech tagging and chunking using conditional random fields and transformation based learning. *In Proceedings Shallow Parsing for South Asian Languages (SPSAL) workshop at IJCAI At Hyderabad India*, 2007.

[34] Luole Qi and Li Chen. A linear-chain crf-based learning approach for web opinion mining. *WISE '10 Proceedings of the 11th international conference on Web information systems engineering*, 2010.

[35] Fam Rashel, Andry Luthfi, Arawinda Dinakaramani, and Ruli Manurung. Building an indonesian rule-based part-of-speech tagger. *International Conference on Asian Language Processing*, 2014.

[36] Katrin Tomanek Roman Klinger. Classical probabilistic models and conditional random fields. *Algorithm Engineering Report*, 2007.

[37] José María Sallán, Joan Baptista Fonollosa, and Albert Sune. *Métodos cuantitativos de organización industrial II.* 2005.

[38] Castrense Savojardo. *MACHINE-LEARNING METHODS FOR STRUCTURE PREDICTION OF B - BARREL MEMBRANE PROTEINS.* PhD thesis, Alma Mater Studiorum - Univerista di Bologna - DOTTORATO DI RICERCA IN INFORMATICA, 2011.

[39] Helmut Schmid. Part of speech tagging with neural networks. *COLING '94 Proceedings of the 15th conference on Computational linguistics - Vol 1*, 1994.

[40] Helmut Schmid. Probabilistic part of speech tagging using decision trees. *International Conference on New Methods in Language Processing*, 1994.

[41] Helmut Schmid. Improvements in part-of-speech tagging with an application to german. *ACL SIGDAT-Workshop*, 1995.

[42] Tej Bahadur Shahi, Tank Nath Dhamala, and Bikash Balami. Support vector machines based part of speech tagging for nepali text. *International Journal of Computer Applications Volume 70*, 2013.

[43] Prajadhip Shina, Nuzotalu M Veyie, and Bipul Syam Purkayastha. Enhancing the performance of part of speech tagging of nepali language through hybrid aproach. *International Journal of Emerging Technology and Advanced Engineering*, 2015.

[44] Evgeny Stanilovsky and Llu'is Padr'o. Freeling 3.0: Towards wider multilinguality. *Language Resources and Evaluation Conference (LREC 2012) ELRA*, 2012.

[45] M Taulé, Martí M.A., and Recasens M. Ancora: Multilevel annotated corpora for catalan and spanish. *Proceedings of 6th International Conference on language Resources and Evaluation*, 2009.

[46] Hanna Wallach. Conditional random fields: An introduction. *University of Pennsylvania CIS Technical Report MS-CIS-04-21*, 2004.

[47] Majoros WH. Conditional random fields. online supplement to: Methods for computational gene prediction. *Cambridge University Press*, 2007.

[48] Phan Xuan-Hieu, Nguyen Le-Minh, Inoguchi Yasushi, and Horiguchi Susumu. High-performance training of conditional random fields for large-scale appli-

cations of labeling sequence data. *IEICE Transactions on Information and Systems E90D(1)*, 2007.

[49] Archit Yajnik. Part of speech tagging using statistical approach for nepali text. *International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol 11*, 2017.