



**UNIVERSITAT POLITÈCNICA DE VALÈNCIA**  
Departamento de Sistemas Informáticos y Computación

**Evaluacion de etiquetadores morfosintacticos**  
**Lingüística Computacional**

**Luis Cardoza Bird.**

**Octubre de 2023**

# Tarea 1

*Evaluación del etiquetador ‘hmm’ sobre el corpus ‘cess-esp’ utilizando el juego de categorías completo y reducido.*

Utilizando el etiquetador hmm basado en modelos de Markov, se realizará una validación cruzada sobre 10 particiones del corpus. Barajar el corpus antes de realizar las particiones. Presentar los resultados en forma de tabla y gráficamente, incluyendo los intervalos de confianza.

---

Fold	Original		Reduced	
	Accuracy	C. Interval	Accuracy	C. Interval
0	0.902139	±0.001398	0.931018	±0.001204
1	0.895361	±0.001443	0.924278	±0.00126
2	0.893981	±0.001448	0.923809	±0.001261
3	0.890366	±0.001469	0.921614	±0.001277
4	0.893	±0.001455	0.920461	±0.001287
5	0.898486	±0.001421	0.92565	±0.001247
6	0.898866	±0.001418	0.928411	±0.001225
7	0.89653	±0.001433	0.926897	±0.001237
8	0.89688	±0.001431	0.92503	±0.001252
9	0.899765	±0.001415	0.927794	±0.001232

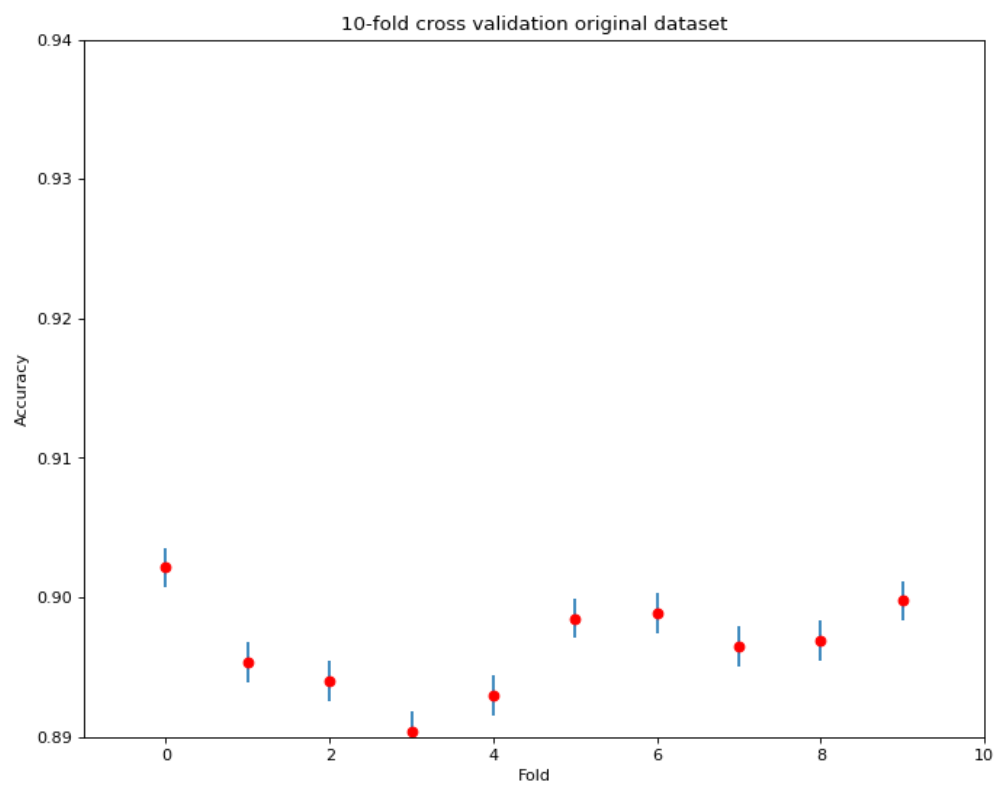


Figura 1.1 Precisiones e intervalos de confianza con corpus original

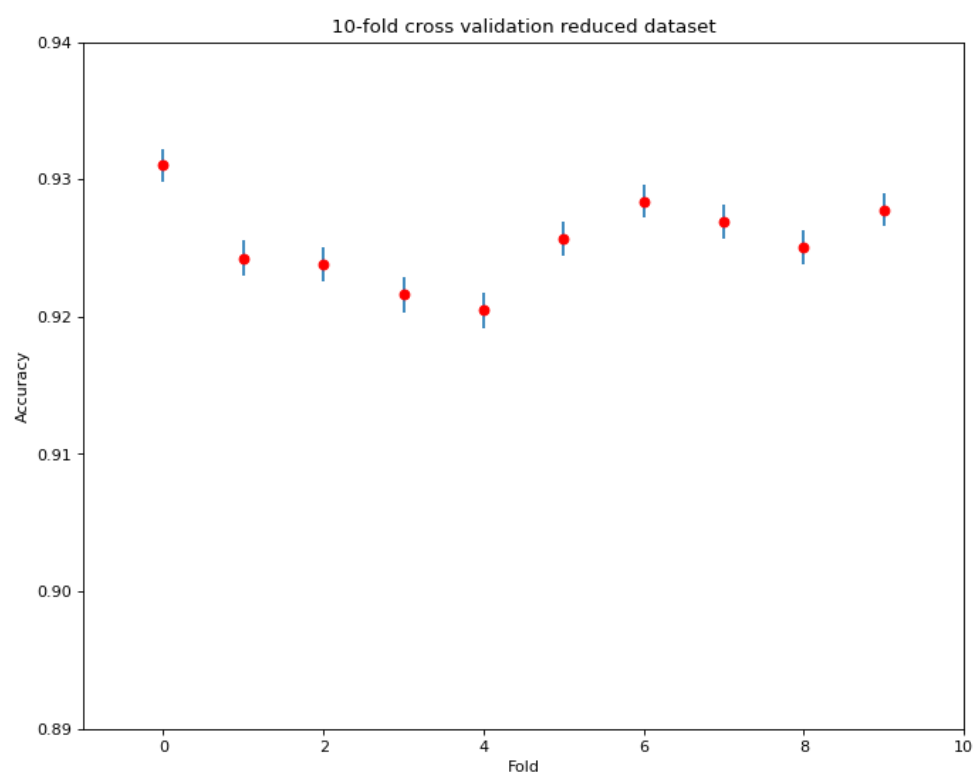


Figura 1.2 Precisiones e intervalos de confianza con corpus de categorías reducidas

Tras descargar el corpus 'cess-esp' lo preparamos para su análisis guardando una copia original y creando otra con categorías reducidas. Esto se consigue recortando todas las etiquetas para que su longitud sea como máximo igual a dos por defecto, salvo los verbos (v) y los signos de puntuación (F) que pueden ser de tres. Además, se eliminan las anotaciones de la forma: (u'\*0\*', u'sn').

Una vez están listos los datos, realizamos con los dos corpus que tenemos, una validación cruzada de diez particiones sobre el método de Modelos Ocultos de Markov. Es reseñable la clara superioridad por parte de los resultados obtenidos con el juego de categorías reducidas, lo cual se debe a una menor variedad de etiquetas que hace la tarea de clasificación más sencilla.

## Tarea 2

*Evaluación de las prestaciones del etiquetador respecto a la cantidad de datos de aprendizaje.*

Se trata de estudiar cómo varían las prestaciones del etiquetador hmm cuando varía el tamaño del corpus de aprendizaje. Para este experimento se dividirá el corpus de entrenamiento en 10 partes de tamaño similar. La partición 10 se tomará como test, y las 9 particiones restantes se tomarán como entrenamiento. En cada ejecución, se irá incrementando sucesivamente el tamaño del corpus de entrenamiento, manteniendo la partición de test. Importante: Para esta tarea no es necesario realizar la validación cruzada.

---

Training Partitions	Accuracy	IC
1	0.7863771	$\pm 0.0058098$
2	0.8292546	$\pm 0.0037329$
3	0.8509738	$\pm 0.0028956$
4	0.8656062	$\pm 0.002407$
5	0.8761599	$\pm 0.0020791$
6	0.8842154	$\pm 0.0018446$
7	0.8903844	$\pm 0.0016687$
8	0.8953809	$\pm 0.0015294$
9	0.8997655	$\pm 0.0014149$

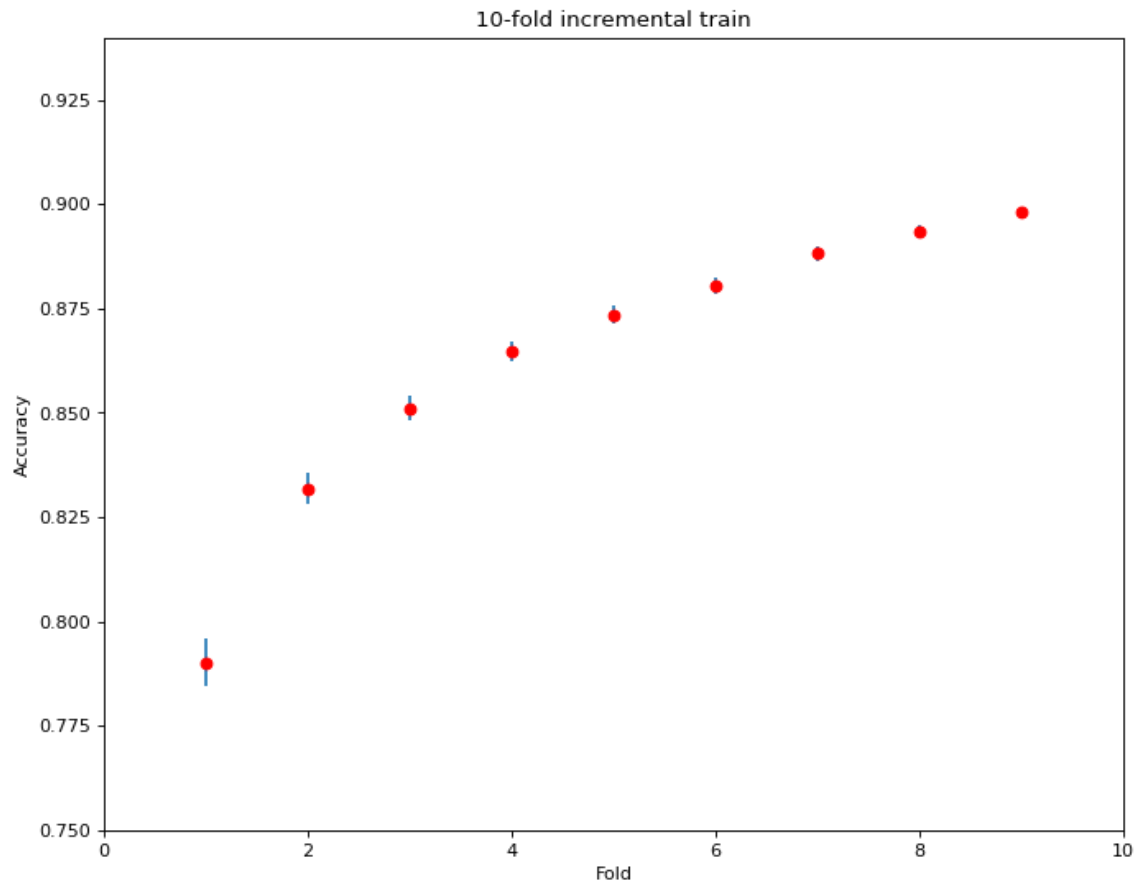


Figura 2.1 Precisiones e intervalos de confianza de entrenamiento incremental con el corpus original

Estableciendo la décima partición como partición de test y guardando el resto para el entrenamiento incremental, al trabajar sobre el corpus original observamos cómo la precisión aumenta a la par que el conjunto de entrenamiento, con un desarrollo que recuerda a una función logarítmica. Este resultado era predecible, puesto que cuanto mayor sea el conjunto de datos de entrenamiento, más precisión se espera de un etiquetador. No obstante, si la partición de test resulta demasiado pequeña no podríamos ofrecer un intervalo de confianza lo suficientemente fiable, además de enfrentarnos al riesgo de incurrir en un sobreaprendizaje.

# Tarea 3

## *Evaluación del método de suavizado para palabras desconocidas para el etiquetador tnt.*

El etiquetador tnt por defecto no incorpora un método de suavizado para las palabras desconocidas. Utiliza un método basado en los sufijos de las palabras para construir un modelo para las palabras desconocidas (Affix Tagger). En base al sufijo de la palabra desconocida le asigna una categoría morfosintáctica. Este método funciona razonablemente bien para el inglés. En concreto, se trata de estudiar diferentes longitudes del sufijo (número de letras que se tienen en cuenta) y estudiar cómo varían las prestaciones del etiquetador. Una vez se haya decidido el sufijo que mejores prestaciones proporciona, incorporarlo como modelo de suavizado al etiquetador tnt y comprobar si aumenta sus prestaciones.

---

Precisión media para sufijos de distinta longitud	
Tamaño	Precisión
1	0.2447989510124732
2	0.2825037815165259
3	0.2955298964139259
4	0.26491860032408826
5	0.225349226446894

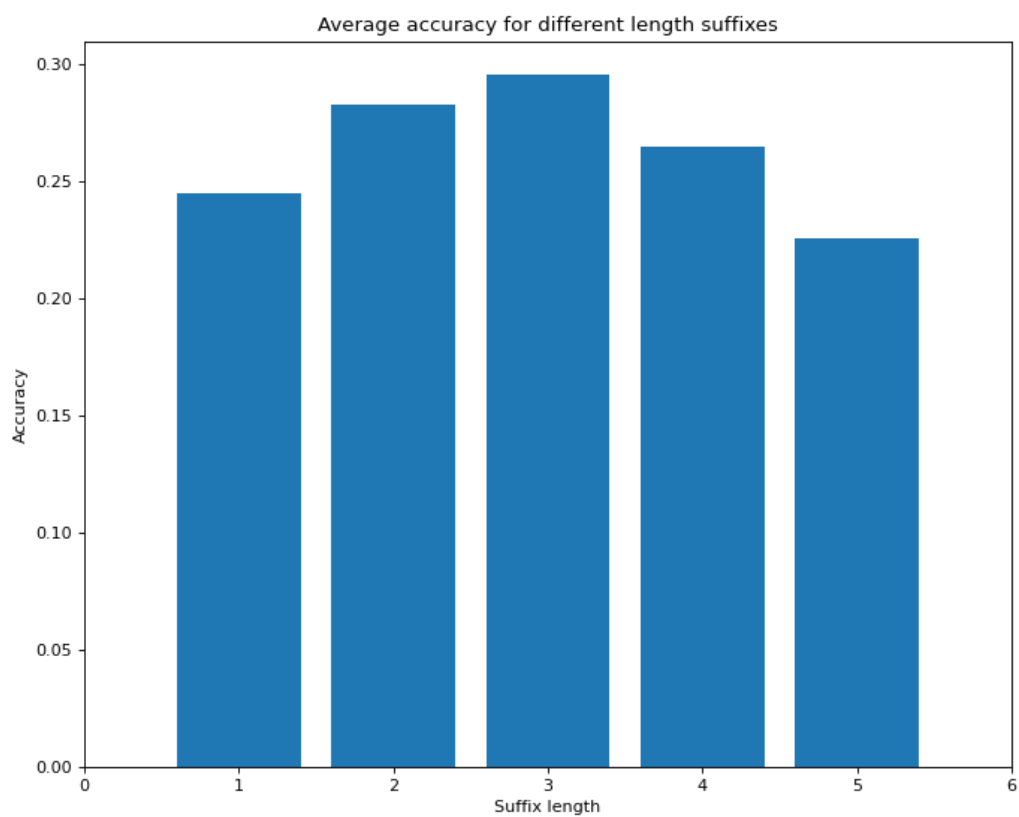


Figura 3.1 Precisión media para distintos tamaños de sufijos con el corpus reducido

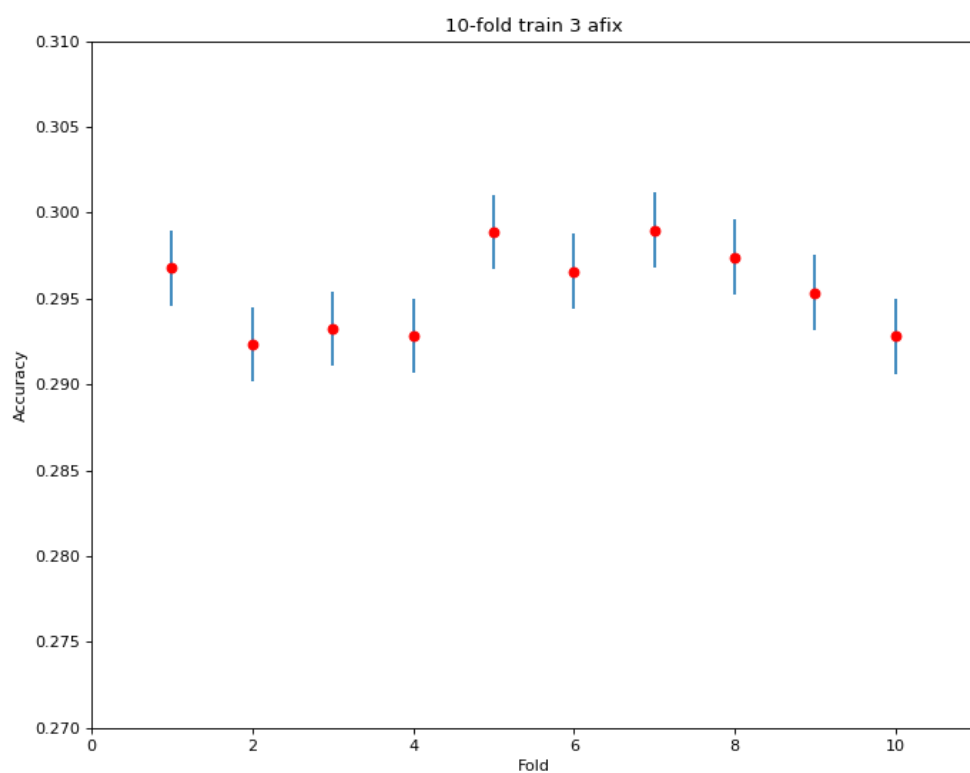


Figura 3.2 Precisiones e intervalos de confianza para sufijo de longitud tres



Trabajando con el corpus de categorías reducidas de nuevo, nos disponemos a efectuar un suavizado para las palabras desconocidas a través de la implementación de un modelo cuyo entrenamiento se base en los sufijos de las palabras. Para ello se le especifica una longitud de sufijo con la que trabajar, eligiendo en nuestro caso longitudes de entre 1 y 5 caracteres. Posteriormente puede verse claramente que los sufijos de tamaño tres ofrecen los mejores resultados.

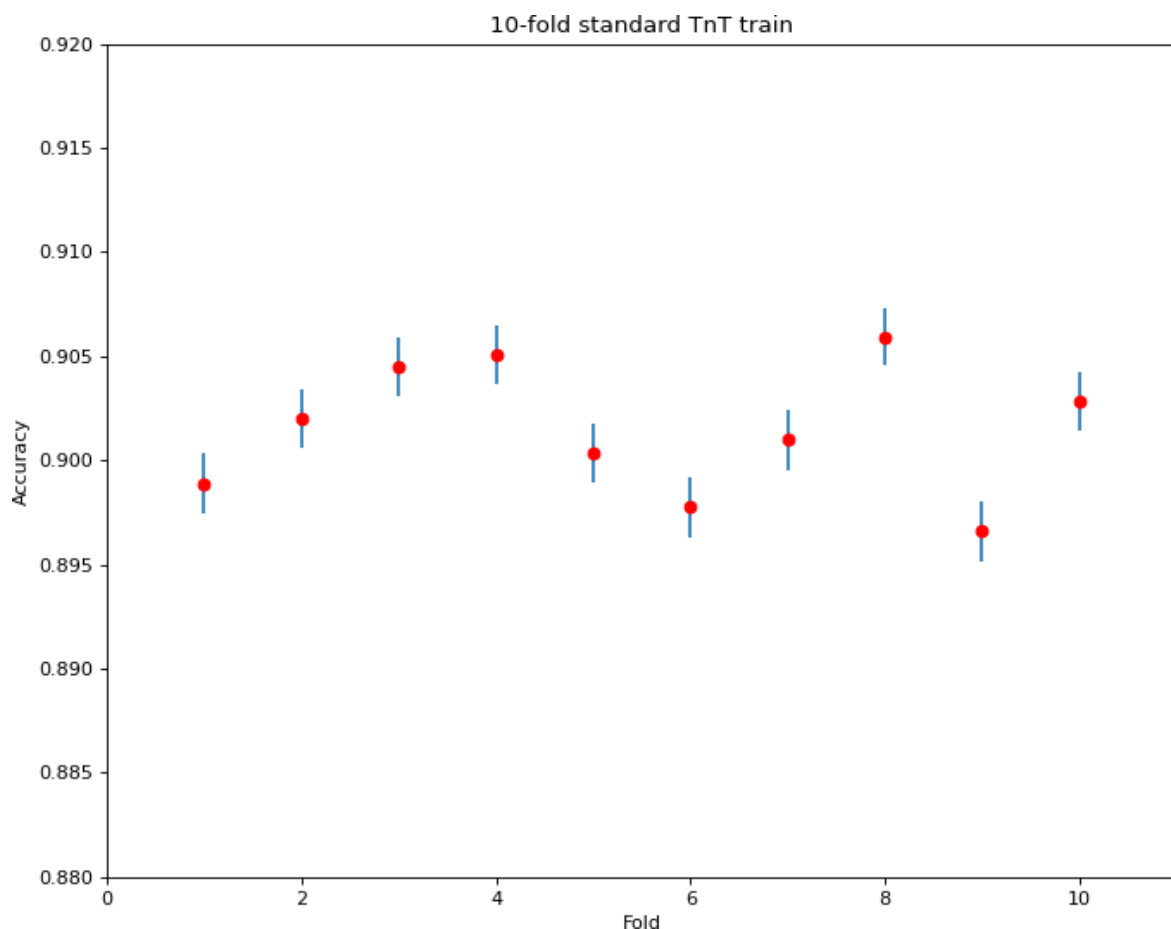


Figura 3.3 Precisiones e intervalos de confianza para TnT sin suavizado

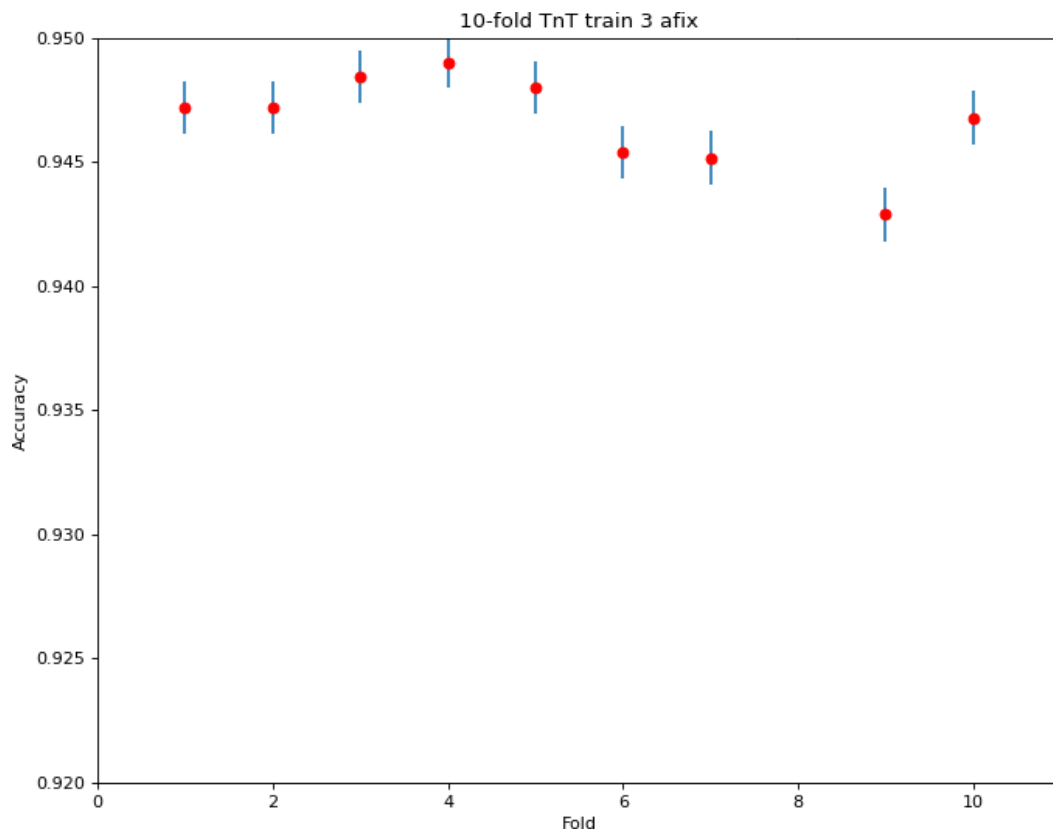


Figura 3.4 Precisiones e intervalos de confianza para TnT suavizado con sufijos de longitud tres.

Tras implementar el suavizado con sufijos de longitud tres al etiquetador TnT observamos una notable mejoría en los resultados obtenidos con respecto a la ausencia de suavizado.

# Tarea 4

## *Evaluación del resto de etiquetadores.*

Se deberán utilizar otros paradigmas de etiquetado. Como mínimo el etiquetador de Brill y algún otro como, CRF, perceptron. Se deberá realizar una comparativa de prestaciones respecto a los etiquetadores tnt y hmm, utilizando el juego de categorías reducido. Cuando se utilice el etiquetador de Brill, probar con diferentes etiquetados iniciales, por ejemplo probar con Unigram Tagger y con hmm tagger. La comparación puede ser sólo de una partición, si el coste temporal de la validación cruzada requiere mucho tiempo.

---

Clasificador	Precisión	IC
Brill con unigramas	0.9009569628196412	$\pm 0.0014213621340826866$
Brill con bigramas	0.7690738900800084	$\pm 0.002005216627315104$
Brill con trigramas	0.7480520838780526	$\pm 0.002065675256883218$
Brill con HMM	0.9289860377555823	$\pm 0.001222129449747496$
CRF	0.9574857501438059	$\pm 0.0009600064777857705$
Perceptrón	0.9676828949432621	$\pm 0.0008414406730028147$
HMM	0.9259007477906186	$\pm 0.0012463208876931171$
TNT	0.9028395126287716	$\pm 0.0014092591547456078$

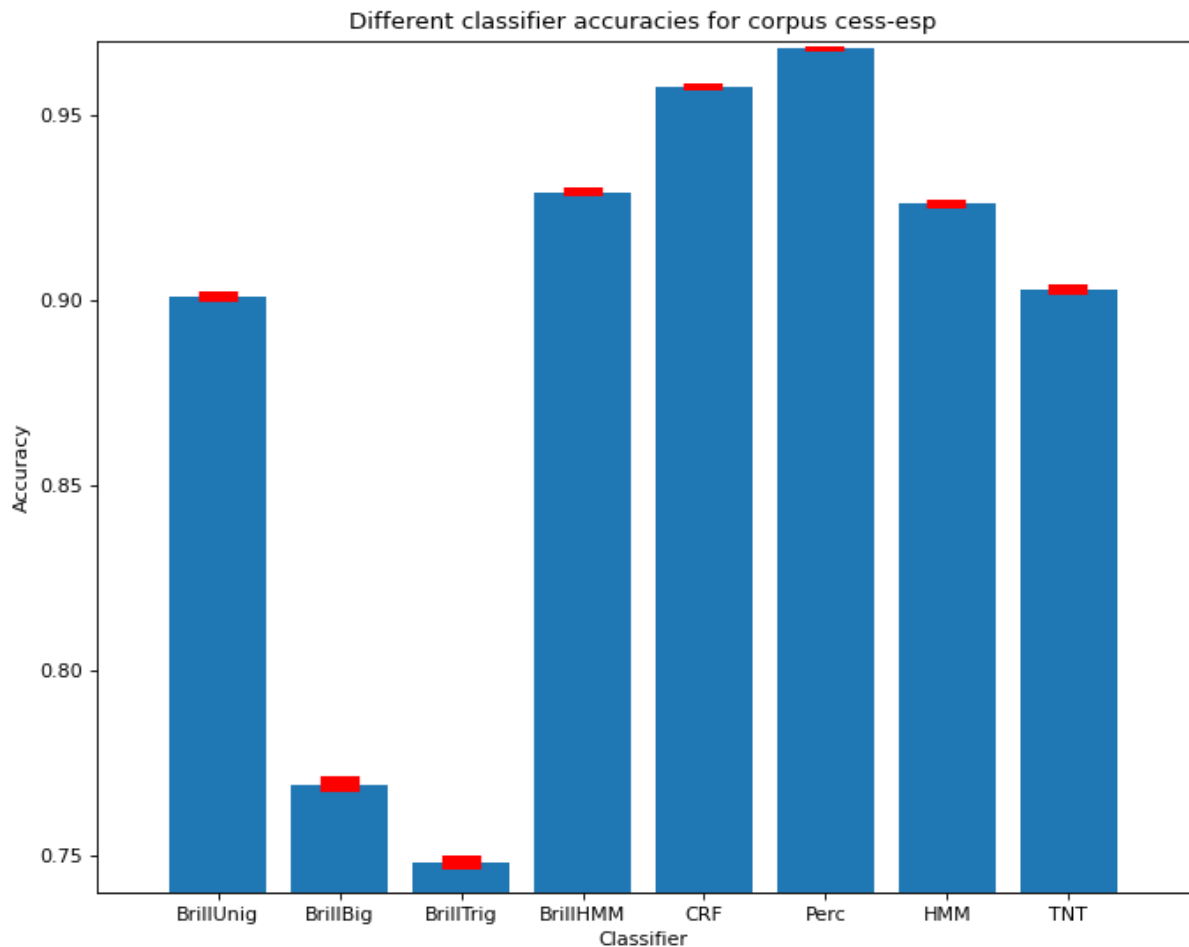


Figura 4.1 Precisiones e intervalos de confianza para distintos clasificadores

En esta tarea expandimos el estudio a diversos etiquetadores del paquete NLTK y los comparamos con HMM y TNT. Primero comenzamos con las distintas variaciones del etiquetador de Brill, es decir, utilizando Brill a la vez que se realiza un pre-etiquetado con distintos etiquetadores. Inicialmente se utilizan 3 modelos de n-gramas, de los que sale por delante el de unigramas, aunque ninguno de ellos supera los etiquetados con HMM y TNT. De hecho, el pre-etiquetado con HMM es el único que permite a Brill superar la barrera del HMM convencional.

Finalmente al trabajar con CRF y Perceptrón obtenemos resultados aún mejores, quedando clara la superioridad del segundo con una precisión cercana al 97%. Esto se debe a que ambos son algoritmos de separación lineal de datos, algo que es posible en bastantes idiomas, entre los que se encuentra el español.

# Tarea 5

## *Evaluación de los paquetes Spacy/Stanza*

Realizar un estudio de la herramienta Spacy y Stanza. Considerar diferentes aspectos: facilidad/problemas de instalación, facilidad de uso, documentación, funcionalidad, etc.

Usar Spacy y Stanza para realizar el etiquetado morfosintáctico del texto del fichero Alicia.txt. Se debe entregar un fichero tipo texto con el formato: palabra/etiqueta.

---

### - **Spacy**

Los mayores problemas que ofreció el paquete Spacy fue el manejo de sus modelos, ya que en la documentación no existe referencia a descargar modelos de forma independiente y al momento de ejecutar el código, siempre existía un error en esta línea: `nlp = spacy.load("es_core_news_sm")`

Por suerte, la solución estaba en un foro donde explican que cada modelo se debe descargar y una vez hecho ese paso, se pudo proseguir con la ejecución.

Las funciones principales del paquete son:

- Tokenización
- Lematización
- Etiquetado morfosintáctico (POS tagging)
- Reconocimiento de entidades nombradas (NER)
- Análisis de dependencias
- Identificación de oraciones
- Soporte para múltiples idiomas

Cabe destacar que la mayor parte de la experiencia con este paquete se obtiene al utilizarlo como librería en otros lenguajes de programación.

Resaltar también su documentación, la cual incluye ejemplos que, en muchos casos, resultan útiles para aprender el uso de la herramienta más rápidamente. Lo que se agradece pues no es precisamente intuitiva.

## - **Stanza**

Las principales funciones que Stanza ofrecia son:

- Tokenización
- Lematización
- Etiquetado morfosintáctico (POS tagging)
- Segmentación morfológica (en algunos idiomas)
- Soporte para múltiples idiomas
- Interfaz con CoreNLP

Durante el desarrollo del ejercicio utilizando stanza, se encontraron diversos problemas, que dificultaron el desarrollo, siendo el principal que el tiempo de ejecución es mas lento en comparacion a Spacy, adicionalmente los modelos que ofrecen son muy limitados y requiere de una cantidad de recursos computacionales para poder funcionar adecuadamente.

## TAREA 6

Entrenar el *Spacy* con el corpus *cess-esp* para realizar *POS tagging* en español y obtener las prestaciones del etiquetador en términos de *accuracy*.

Utilizar spaCy con el corpus CESS-ESP presentó varios desafíos y consideraciones:

- **Formato de Datos:** El corpus CESS-ESP, disponible a través de NLTK, tiene su propio formato de etiquetado. Convertirlo al formato requerido por spaCy fue un paso esencial.
- **Evolución de spaCy:** Las versiones más recientes de spaCy (3.x) han introducido cambios significativos en la forma de definir y entrenar modelos, lo que requirió adaptaciones en el código.
- **Entrenamiento:** Aunque spaCy facilita el entrenamiento de modelos, es esencial ajustar correctamente los parámetros y procesar los datos de entrada para obtener resultados óptimos.
- **Evaluación:** La precisión inicialmente reportada fue incorrecta debido a problemas en la comparación de etiquetas predichas y reales. Corregir esto fue crucial para obtener una métrica de rendimiento válida.

**Accuracy: 0.6009:** El modelo logró predecir en un 60.01% la etiqueta morfosintáctica correcta dentro del modelo CESS-ESP