Facultad de Ingeniería
Instituto de Computación

Universidad de la República

Resolución de Correferencias Entre Fuentes de Opiniones

Estudiante: Mauricio Carbajal

Tutora: Aiala Rosá

2015

Resumen

Con el objetivo de mejorar un sistema de búsqueda de opiniones, se presenta el diseño y la implementación de un sistema para la de resolución de correferencias entre fuentes de opiniones, para el idioma español.

El módulo implementado toma distintas ideas de estos nuevos enfoques, buscando integrarlas en una arquitectura modularizada, separando en todo momento la información brindada por aplicaciones de terceros del módulo concreto que resuelve el problema planteado.

Para la resolución de correferencias se utiliza un enfoque heurístico, tomando como punto de partida un proyecto de grado existente, desarrollado en nuestra facultad en el año 2010.

Los resultados alcanzados son de **74%** para la recuperación de fuentes, y de **85%** para la resolución de correferencias[1].

[1] Resultados calculados utilizando la medida F

Contenido

Capítulo 1

Introducción

Según mediciones realizadas por Data Media[1] a fines del año 2011, en Uruguay la lectura de prensa igualó a la lectura en papel. Esto muestra que la tendencia es a que la mayoría de las personas se informen a través de medios digitales.

Dejando de lado la practicidad y los beneficios ambientales, una enorme ventaja de los medios digitales es la posibilidad de utilizar herramientas para personalizar la información a consumir según los intereses particulares de cada usuario.

En el contexto de estas herramientas de búsqueda personalizada, es que surge la necesidad de detectar las entidades presentes en un texto y las diversas formas de referirse a ellas.

Tomando el caso concreto de las opiniones en textos, si queremos recuperar las opiniones emitidas por cierta persona, es posible que debido al estilo de redacción se utilicen distintas menciones a la misma persona como forma de evitar la redundancia. Ejemplo:

"Vázquez va a encabezar un gobierno más predecible, más cauteloso y ordenado", dice ***el analista político Gerardo Caetano****.*

"La gran prueba para el Frente Amplio comienza ahora. Debe demostrar que todavía puede tener éxito ahora que la bonanza ha terminado", añade ***Caetano****.*

Podemos ver que las dos menciones en negrita referencian a una misma entidad: el analista político Gerardo Caetano.

Por ello es necesario resolver cuándo dichas menciones refieren a una misma entidad de discurso, para mejorar la recuperación de información.

La motivación principal del estudio de las correferencias en este proyecto, en contexto de búsquedas de opiniones por fuente y asunto, es poder recuperar una mayor cantidad de opiniones que estuvieran asociadas a cierta fuente que no es exactamente la que buscó el usuario, pero sí es correferente con ésta.

1.1 Objetivos

Los objetivos específicos fueron definidos tras los primeros meses de transcurso de este proyecto, que fueron dedicados a evaluar las herramientas existentes y los cambios necesarios para integrarlas al sistema buscOpiniones[4].

A continuación se presentan los objetivos que se decidieron para este proyecto.

1.1.1 Resolución de correferencias entre fuentes de opinión.

Se consideró integrar el módulo de correferencias desarrollado en 2010 [3] (tarea que no había sido posible realizar en el proyecto buscOpiniones) y luego hacer sobre dicho desarrollo las mejoras necesarias.

Si bien el funcionamiento del módulo en esencia era el esperado, no contaba con un manejo de errores suficiente para integrarlo en una cadena de procesamiento. Esto sumado a que actualmente FreeLing ofrece algunas funcionalidades de interés para la resolución de este problema, y sumado a la idea de emplear un parser de dependencias, hizo que se definiera como primer objetivo desarrollar un nuevo módulo para la resolución de correferencias entre fuentes de opiniones.

Se espera que dicho módulo mejore las heurísticas definidas por el proyecto anterior, incorporando la información de nombres de persona u organización provista por la nueva versión de FreeLing, así como el uso de un parser de dependencias.

1.1.2 Integración de información de distintos recursos.

Al iniciar el proyecto, se notó que tanto el módulo de correferencias desarrollado en 2010, como el identificador de fuentes de opiniones, como FreeLing, necesitaban de la intervención del usuario para procesar un texto y cada uno brindaba su salida en un archivo, de diverso formato.

Fue por eso que se fijó como segundo objetivo el integrar todas las herramientas en una sola tarea, con el fin de uniformizar la información en una base de datos y permitir eventualmente integrar nuevos recursos de información sin mayores dificultades.

Capítulo 2

Estudios Previos

En este capítulo se hará una breve reseña de algunos trabajos relacionados al problema de la resolución de correferencias. El marco teórico parte del realizado en el proyecto anterior[3], e incorpora algunos trabajos más recientes, que si bien están basados en aprendizaje automático, aportan mucho a las ideas que guían este proyecto de grado.

2.1 Conceptos Importantes

En base a lo expuesto en [2] y mencionado en [3] se hará a continuación un breve repaso de algunos de los conceptos involucrados en la resolución de correferencias: referencia, correferencia, anáfora, elipsis y otros tipos de expresiones referenciales de interés.

2.1.1 Referencia

Hablamos de una *referencia* cuando se utiliza una expresión para hacer mención a una entidad del mundo real (por ejemplo mención a una persona, o a una organización). A esta expresión, se la denomina *expresión referencial* y se denomina *referente* a la entidad referenciada por la expresión.
A la tarea de encontrar las entidades referenciadas por expresiones referenciales, se le denomina *resolución de referencias*.

2.1.2 Anáfora

Hablamos de una *anáfora* cuando hacemos una referencia a una entidad introducida anteriormente en el discurso (a la cual llamamos *antecedente*). Debe ser posible deducir mediante restricciones sintácticas el antecedente al cual nos estamos refiriendo, ya que las referencias anafóricas están atadas al contexto por ser semánticamente vacías.
Un ejemplo:

"Ayer llegaron Luis y Ana. Él *está muy contento de haber vuelto"*

Como se puede notar, la resolución de la anáfora está ligada al contexto, y también a restricciones sintácticas sobre el elemento anafórico. Sabemos que el pronombre "Él" solo puede estar refiriéndose a Luis, y no a Ana.

2.1.3 Elipsis

Hablamos de una *elipsis* cuando en el discurso se omite la mención de alguna entidad debido a que es posible inferirla por contexto.

Un ejemplo de **elipsis verbal** (omitiendo un verbo de la oración):

"Las mujeres traen la bebida y los hombres (traen) la comida"

Un ejemplo de **elipsis nominal** (omitiendo un sujeto u objeto de la oración):

"(yo) He desayunado esta mañana."

Es interesante notar que la posibilidad de omitir enteramente el sujeto de la oración[17] no está disponible en todos los idiomas. Por ejemplo en inglés, (*"**I** had breakfast this morning"*), no es posible crear una oración gramaticalmente correcta sin utilizar un sujeto. Las lenguas que sí permiten la posibilidad de omitir el sujeto son llamadas

lenguas *pro-drop*.

2.1.4 Aposición

Hablamos de una *aposición* cuando en un discurso se utiliza un grupo nominal que contiene la unión de dos elementos gramaticales distintos. Se utiliza típicamente para introducir una entidad del mundo real al discurso.
Un ejemplo de **Aposición**:

"El titular de Transporte y Obras Públicas, Victor Rossi"

2.1.5 Otros tipos de expresiones referenciales

Existen varios tipos de expresiones referenciales:

Frases nominales indefinidas

Son aquellas donde el especificador es un determinante indefinido (un, una, algún).
"Según informaron **algunos** medios."

Frases nominales definidas

Son aquellas donde la entidad referida es identificable por el lector, ya sea por una mención previa, o porque el lector identifica esa entidad en el mundo real.
"Según informó **el** presidente."

Uso de pronombres

Como se explicó en la sección de Anáforas, los pronombres pueden ser utilizados como expresiones referenciales, y son una forma de referencia definida.
"Su hermana **le** dejó las llaves en la puerta."

Uso de nombres propios

Los nombres propios son la forma más común de referenciar una entidad del mundo real. Son usados tanto para introducir nuevas entidades al discurso (en las primeras menciones), así como para referir a entidades ya mencionadas.
"Según informó **El Observador**."

"**Tabaré Vázquez** realizó un comunicado de prensa. **Vázquez** informó que (...)."

2.1.6 Correferencias

Decimos que dos expresiones referenciales son *correferentes* cuando ambas hacen referencia a una misma entidad del mundo real.

Tomando el ejemplo mostrado en [3]:

*"La relación de **Diego Forlán** con la hinchada del Atlético de Madrid y la prensa española debe ser única en el mundo. Cualquier otro equipo se hubiera rendido a los pies de un jugador con la mitad de los logros de **Forlán**, sin mencionar que si **el delantero** fuera argentino o brasileño, probablemente el periodismo hubiera dedicado el triple de páginas y elogios a **su figura**, que guarda un perfil bajo inusitado."*

En este fragmento de texto existen cuatro expresiones referenciales que mencionan a una misma entidad: Diego Forlán. Más adelante se analizaran varios tipos de correferencia, incluyendo las utilizadas en este texto.

2.1.7 Tipos de Correferencia

Los tipos de correferencia se clasifican en función de la relación existente entre la expresión referencial y su antecedente:

Identidad

Cuando existe una correferencia de tipo identidad, significa que es posible intercambiar la expresión referencial por su antecedente, sin alterar el significado de las oraciones. Un ejemplo de esto, es el uso de sinónimos, o hiperónimos:

"*El presidente Vázquez* emitió un comunicado de prensa (...).
El *mandatario* explicó (...)"

Donde vemos que la relación entre la expresión referencial (mandatario) y su antecedente (El presidente Vázquez) es de tipo hiperónimo/hipónimo, ya que "mandatario" es un tipo más general que "presidente".

Conjunto - Miembro

Cuando existe una correferencia de tipo Conjunto - Miembro, significa que el antecedente es un conjunto, del cual es parte la expresión referencial, o viceversa. Ejemplo:

"*Presidencia* emitió un comunicado de prensa (...).
El *portavoz* explicó (...)"

Conjunto - Subconjunto

Es un caso similar al anterior:

"*Un grupo de mujeres* reclamó hasta altas horas de la tarde. *Algunas mujeres* pensaban retomar mañana temprano"

Parte de

En este caso, la expresión referencial alude a una parte del antecedente:

"Ayer llevé el *auto* al taller. El *motor* no tenía arreglo"

Este último caso no es frecuente en el contexto de opiniones.

En el listado anterior se encuentran los tipos de referencia más frecuentes en el idioma español. En la siguiente sección, se hará una revisión sobre trabajos existentes relacionados al problema de resolver correferencias.

2.2 Trabajos relacionados

A continuación se presentan algunos trabajos relacionados a la resolución de correferencias, tomando como base el trabajo realizado por Vincent Ng[5], donde se exponen los distintos enfoques con los que se ha tratado la resolución de correferencias. Luego se presentará un breve resumen de los trabajos presentados al Semeval de correferencias (2010), en particular aquellos que buscan resolver el problema para el idioma español.

2.2.1 Investigación sobre resolución supervisada de correferencias : los primeros 15 años

Introducción

Este artículo realizado por Vincent Ng, titulado originalmente como "Supervised Noun Phrase Coreference Research : The First Fifteen Years" [5] realiza una clasificación de las distintas ideas que han surgido en el campo de la resolución de correferencias, evaluando los distintos aspectos del problema.

Debido a lo completo y extensivo de este compendio (donde se citan más de 90 trabajos distintos), el siguiente resumen pretende exponer las ideas y los modelos esenciales utilizados al resolver correferencias.
En este artículo se considera el problema de resolver correferencias como el de encontrar en un texto cuales grupos nominales (NP) refieren a la misma entidad del mundo real, tarea relacionada con resolver cuáles son los elementos anafóricos[1] del el texto.
Otra restricción importante es que en este artículo se evalúan únicamente las formas de resolver correferencias de tipo *identidad*, dejando de lado otros tipos de correferencia (más adelante en este informe se mencionan varios tipos de correferencia).

[1]En este artículo se consideran elementos anafóricos todos aquellos NPs que deben ser resueltos. Esto incluye además de los elementos clásicos de las anáforas lingüisticas, nombres comunes que pueden estar referenciando a una entidad presentada anteriormente.

El autor afirma que los enfoques del problema se han ido moviendo desde enfoques heurísticos hacia enfoques que utilizan aprendizaje automático.

Corpus anotados con correferencias

En primer lugar se mencionan varios conjuntos de datos anotados con correferencias para el idioma inglés, argumentando que esta abundancia de material puede ser una de las causas por las que el enfoque hacia el problema se haya vuelto más probabilístico y cercano al aprendizaje automático.
Cabe notar que el único corpus mencionado para el idioma español es el AnCora Corpus[20].

Modelos para pares de menciones

Estos modelos evalúan si dos grupos nominales son correferentes entre sí. En este tipo de modelo, la transitividad de las correferencias no queda implementada, por lo que requieren de un mecanismo posterior de agrupamiento en clases (*Clustering*).

Uno de los principales problemas que presenta es que los datos de entrenamiento consisten en pares de grupos nominales, y dado que en un texto pueden presentarse muchos grupos nominales donde la gran mayoría de ellos no son correferentes, las instancias negativas de entrenamiento son demasiadas, por lo que las clases de grupos nominales correferentes resultan sesgadas.

Una forma de resolver esto, es especificar la forma con la que se crean las instancias de entrenamiento, buscando balancear las instancias positivas con las instancias negativas.

Entrenamiento

En primera instancia, en [21] se crean instancias positivas de entrenamiento entre el grupo nominal NP_k y el candidato más cercano NP_j. Además, crea instancias negativas de entrenamiento en las palabras del medio (NP_{j+1} ... NP_{k-1}).

Luego se proponen varias mejoras como únicamente vincular instancias positivas entre NPs que no sean pro-nominales (ya que no aporta entrenar correferencias entre

grupos nominales y pronombres), o únicamente vincular instancias positivas entre NPs que concuerden en número y género.

Clasificadores utilizados

El tipo de clasificador más utilizado es el de tipo árbol de decisión. Últimamente también se han utilizado otros modelos que brindan un valor de confianza, por ejemplo modelos de máxima entropía, o el modelo SVN (*Support Vector Machines*)

Agrupamiento en clases (*Clustering*)

Respecto al agrupamiento en clases de grupos nominales correferentes, el modelo más básico consiste en elegir el primer antecedente que supere un cierto umbral de correferencia (*closest-first*). Otra forma es seguir buscando y elegir el candidato con mayor probabilidad de ser correferente (*best-first*).

Uno de los problemas de estos enfoques es que existe la posibilidad de que tres grupos nominales terminen en la misma clase de correferencia, aún habiendo certeza de que dos grupos nominales no son correferentes. (Ejemplo: "Mr. Clinton", "Clinton", "she").

Un enfoque más completo para el problema de agrupamiento es considerar el resultado como un grafo ponderado, donde los nodos son los grupos nominales, y las ponderaciones la probabilidad de que sean correferentes. Luego emplear algoritmos de partición de grafos, de forma que conserven conectados los nodos más ponderados.

También se describen otros algoritmos para el agrupamiento, como el basado en árboles de Bell (Luo et al's, 2004).

Desventajas de los modelos de pares de menciones
Existen dos desventajas citadas frecuentemente:

- Al evaluar las menciones de a pares, se sabe qué tan bueno es un candidato para cierto elemento anafórico, pero se desconoce qué tan bueno es respecto a las otras opciones que hay disponibles. Para esto es que surgen los modelos de pares Entidad-mención, a describirse a continuación.

- La expresividad del modelo puede ser baja, en el sentido de que la información extraída de dos grupos nominales no es suficiente para marcarlos como correferentes, especialmente cuando no se tiene información de número o género (o para los pronombres, donde no se tiene ningún tipo de información semántica).

Modelos par entidad-mención

Lo que diferencia a estos modelos de los anteriores, es que cada grupo nominal es asociado con una clase (que contiene varios grupos nominales ya resueltos como correferentes), en lugar de asociar a otro grupo nominal. Esto permite mejorar el problema de expresividad descrito, ya que la clase brinda mayor cantidad de información de la entidad a referenciar.

Utilizando este tipo de modelo, el ejemplo presentado anteriormente sobre los Clintons tendría mejores posibilidades de ser resuelto correctamente, ya que en una primera instancia se agruparía "Mr. Clinton" con "Clinton" y eso constituiría una clase. Luego "she", que por contexto se asocia a "Clinton", también sería evaluada respecto a los otros miembros de la clase a la que se pretende unir, y posiblemente se la descartaría por fallar la concordancia de género.

Para agregar un grupo nominal a una clase, se puede variar el o los atributos restrictivos (ej: concordancia de número), y también se puede variar el criterio para agregar, pudiendo requerir concordancia con todos los elementos de la clase (ALL), o con la mayoría de los elementos de la clase (MORE), o simplemente con algún elemento de la clase (ANY).

Modelos de Puntajes

El modelo entidad-mención mejora la expresividad del modelo clásico de pares de menciones, pero no resuelve el problema de no comparar un candidato respecto a los otros disponibles. Para solucionar esto, es que surgen los modelos de puntajes (*Ranking*).

A cada grupo nominal candidato se le asigna un puntaje, de forma de poder evaluar simultáneamente todos los candidatos disponibles y elegir el de mejor puntaje. De todas formas, este modelo sigue careciendo de la expresividad ya que no realiza la comparación a nivel de entidad (donde podría sumar la información de todas las menciones de la clase).

Uso de fuentes de conocimiento

Otra tarea relacionada al problema de resolver correferencias consiste en computar las características (*features*) de cada grupo nominal. En el artículo se proponen varias características, de las cuales se citan algunas a continuación:

Correspondencia entre cadenas de texto (*match*)

Se propone computar varias funcionalidades relativas a cadenas de texto, tales como correspondencia exacta, sub-cadena, correspondencia entre lemas, distancia de edición entre dos cadenas, frecuencia del término y frecuencia inversa de documento (tf-idf), entre otras.

Características sintácticas

Es posible computar algunas características sintácticas como la distancia de Hobbs[23], que asigna un puntaje a cada candidato a resolver un pronombre, así como computar información obtenida a partir de un árbol de dependencias. También se menciona un trabajo[22] donde se construye una medida para comparar la similitud entre dos árboles de dependencias.

Características gramaticales

También se propone computar las características gramaticales como género, número, categoría gramatical (nombre, verbo, etc.) y tantas otras como sea posible.

Características semánticas

Si bien son más complejas que las anteriores, es posible agregar variables como el sentido de una expresión, utilizando herramientas como Wordnet. Además, se menciona un trabajo[24] que propone medir la distancia semántica entre dos grupos nominales según datos obtenidos de procesar la wikipedia (más precisamente, por ejemplo si un grupo nominal aparece en el primer párrafo del artículo del segundo, éstos están semánticamente relacionados).

Otras consideraciones

El artículo presenta algunas conclusiones respecto al aspecto supervisado del entrenamiento, citando trabajos (ejemplos:[25] [26]) que utilizando aprendizaje no supervisado también han conseguido resultados similares.

También menciona que los trabajos allí presentados han sido desarrollados originalmente para el idioma inglés, aunque algunos han sido probados en otros idiomas, poniendo foco en resolver ciertos tipos de anáfora no presentes en el idioma inglés (este tema será abordado más adelante, en la sección de conceptos involucrados).

Otra de las conclusiones es la dificultad de determinar un estado del arte para la resolución de correferencias, debido a que los investigadores realizan su trabajo en base a distintos conjuntos de datos, utilizando distintas herramientas al pre-procesar, y utilizando diferentes métricas[2] al evaluar sus resultados.

[2]El artículo también trata algunas métricas, pero este tema será abordado por este proyecto en el capítulo "Evaluación"

2.2.2 SemEval 2010 : Correferencias

SemEval es una actividad dónde se evalúan distintos programas relacionados a comprender el significado semántico del lenguaje. Cabe notar que si bien el lenguaje es utilizado intuitivamente por las personas, el análisis computacional del significado ha probado ser una tarea compleja.

En el año 2010, se realizó un SemEval donde se realizaron evaluaciones sobre herramientas que resolvieran correferencias en distintos idiomas. Todas fueron probadas con un mismo juego de datos, y evaluadas segun distintas métricas.
A continuación se estudian aquellos trabajos presentados que resolvían correferencias para el idioma español.

SUCRE

Este trabajo[8] fue uno de los que obtuvo mejores números en las evaluaciones realizadas, pero lo que resulta de mayor interés para este proyecto es el enfoque bajo el cual está implementado.

Su arquitectura se separa en dos etapas: la del pre-procesamiento, donde el texto de entrenamiento es transformado en una base de datos relacional, y la de resolución de correferencias, donde se realiza dicha tarea utilizando la base de datos.

Preprocesamiento
En esta etapa, además de cargar el texto a la base de datos:

- Se procesan y almacenan los atributos existentes a nivel de palabra.
- Se almacenan las marcas que abarquen más de una palabra.

En el modelo de base de datos que se presenta a continuación, se puede comprender mejor la estructura básica de la base de datos.
Consta de tres tablas, una de palabras y sus atributos (Word), otra de marcas sobre palabras (Markable), y otra para representar vínculos entre palabras (Links).

Column	Characteristic
Word Table	
Word-ID	Primary Key
Document-ID	Foreign Key
Paragraph-ID	Foreign Key
Sentence-ID	Foreign Key
Word-String	Attribute
Word-Feature-0	Attribute
Word-Feature-1	Attribute
...	Attribute
Word-Feature-N	Attribute
Markable Table	
Markable-ID	Primary Key
Begin-Word-ID	Foreign Key
End-Word-ID	Foreign Key
Head-Word-ID	Foreign Key
Markable-Feature-0	Attribute
Markable-Feature-1	Attribute
...	Attribute
Markable-Feature-N	Attribute
Links Table	
Link-ID	Primary Key
First-Markable-ID	Foreign Key
Second-Markable-ID	Foreign Key
Coreference-Status	Attribute
Status-Confidence-Level	Attribute

Figure 2.1: Modelo de la base de datos empleado por SUCRE

La tabla de enlaces (*links*) será el lugar donde se insertarán los vínculos entre dos palabras, o grupos de palabras, que sean correferentes.

También es importante notar que en la tabla de palabras no solo se almacena el identificador de palabra, de oración y de documento, sino que también se guardan todos los atributos que luego serán utilizados a la hora de resolver correferencias.

Este enfoque permite centralizar los atributos de palabra y de grupo de palabras en un mismo lugar, independientemente de que dichos atributos sean obtenidos de distintas aplicaciones.

Atributos utilizados

Los atributos a nivel de palabra utilizados por este sistema son: género gramatical (masculino, femenino o neutro), número (singular, plural, o ambos), clase semántica y tipo (ej: pronombre personal, pronombre reflexivo, etc.), rol sintáctico, entre otros.

También se utilizan atributos a nivel de oración, como cantidad de palabras, tipo de oración (simple, compuesta o "compleja"), y también son utilizados algunos atribu-

tos a nivel de documento (si es periodístico, si es un artículo, libro, etc).

Resolución de correferencias

La resolución de correferencias se realiza utilizando un algoritmo supervisado de aprendizaje automático. Se experimentó con cuatro modelos distintos (Árbol de decisión, Naive-Bayes, SVM y máxima entropía), obteniendo los mejores resultados al utilizar Árboles de decisión.

Respecto al agrupamiento en clases (*clustering*), el algoritmo utiliza la política best-fit.

Resultados

Los resultados obtenidos por SUCRE son los siguientes:

Language	ca	de	en	es	it	nl
System	SUCRE (Gold Annotation)					
MD-F1	100	100	100	100	98.4	100
CEAF-F1	68.7	72.9	74.3	69.8	66.0	58.8
MUC-F1	**56.2**	**58.4**	**60.8**	**55.3**	**45.0**	**69.8**
B^3-F1	77.0	81.1	82.4	77.4	**76.8**	**67.0**
BLANC	**63.6**	**66.4**	**70.8**	**64.5**	**56.9**	**65.3**
System	SUCRE (Regular Annotation)					
MD-F1	69.7	**78.4**	**80.7**	70.3	**90.8**	**42.3**
CEAF-F1	47.2	**59.9**	**62.7**	52.9	**61.3**	15.9
MUC-F1	**37.3**	**40.9**	**52.5**	**36.3**	**50.4**	**29.7**
B^3-F1	51.1	**64.3**	**67.1**	55.6	**70.6**	11.7
BLANC	**54.2**	**53.6**	**61.2**	**51.4**	57.7	**46.9**
System	Best Competitor (Gold Annotation)					
MD-F1	100	100	100	100	N/A	N/A
CEAF-F1	**70.5**	**77.7**	**75.6**	66.6	N/A	N/A
MUC-F1	42.5	25.9	33.7	24.7	N/A	N/A
B^3-F1	**79.9**	**85.9**	**84.5**	**78.2**	N/A	N/A
BLANC	59.7	57.4	61.3	55.6	N/A	N/A
System	Best Competitor (Regular Annotation)					
MD-F1	**82.7**	59.2	73.9	**83.1**	55.9	34.7
CEAF-F1	**57.1**	49.5	57.3	**59.3**	45.8	**17.0**
MUC-F1	22.9	15.4	24.6	21.7	42.7	8.3
B^3-F1	**64.6**	50.7	61.3	**66.0**	46.4	**17.0**
BLANC	51.0	44.7	49.3	51.4	**59.6**	32.3

Figure 2.2: Resultados obtenidos por SUCRE y por su mejor competidor según cada métrica. En negrita aparece el mejor resultado obtenido en semeval.

RELAXCOR

Este sistema representa la resolución de correferencias como un problema de partición de grafos. Se crea un grafo que tiene como nodos a todas las menciones y luego, aplicando restricciones sobre las relaciones entre dichos nodos, se particiona el grafo en clases compuestas por menciones correferentes.

El sistema trabaja a partir de las menciones del corpus de prueba, ya que no realiza la tarea previa de detectar las menciones en el texto.
El artículo completo puede consultarse en [9].

TANL-1

Este sistema utiliza aprendizaje automático para la resolución de correferencias. Se utiliza un clasificador binario basado en máxima entropía, con el fin de decidir si existe relacion entre un par de menciones del texto. La detección de menciones está basada analizando la salida de un parser de dependencias.

El artículo completo puede consultarse en [10].

UBIU

Este sistema busca una solución independiente del lenguaje, utilizando aprendizaje basado en memoria (MBL, [29]). Para la resolución, utiliza información sintáctica de cada lenguaje, siendo esta la única adaptación necesaria al integrar un nuevo lenguaje.

El artículo completo puede consultarse en [11].

Capítulo 3

Análisis del Problema

En este capítulo se analiza el problema de resolver correferencias entre fuentes de opiniones, considerando también el problema de resolver los predicados que no tienen fuente, problema que guarda gran similitud con el de la resolución de correferencias.

Primero se expondrá la situación a resolver, para luego presentar la solución que se propone para ambos problemas

3.1 Marco de Trabajo

Este proyecto surge en el contexto de varios desarrollos anteriores. A continuación se realiza una breve descripción de tres sistemas ligados al dominio del problema, para más adelante detallar en el capítulo "Implementación" la forma concreta en que se interactúa con ellos.

También se describen algunos trabajos que han aportado ideas para la solución propuesta por este proyecto.

3.1.1 Módulo identificador de opiniones

Este trabajo [6],[7] fue desarrollado en la Facultad de Ingeniería, como tesis de doctorado.

En él se estudian las distintas expresiones utilizadas para reproducir una opinión en el idioma español, y a partir de dicho estudio se crea un modelo basado en reglas que permiten identificar fuente, predicado, asunto y mensaje de la opinión. A continuación se muestra un ejemplo de una opinión detectada por el sistema:

Consultado sobre *la lentitud de los procesos judiciales uruguayos*, Carranza **respondió**: *"Hay una situación de un muy alto número de presos sin condena, hay que agilizar los procesos"*.

En este caso, "respondió" es el predicado de opinión, y "Carranza" la fuente asociada. El asunto es "la lentitud de ... uruguayos", y el mensaje de opinion es lo citado entre comillas.

Este sistema basado en reglas contextuales, obtiene valores de medida F parcial[1] de **92%** para el reconocimiento de predicados, **81%** para fuentes, **75%** para asunto, **89%** para el mensaje de opinión, y **85%** para la opinión completa. La medida F

[1]Medida obtenida considerando también el reconocimiento parcial de elementos

exacta [2] para reconocimiento de fuentes es de **79%**.

3.1.2 Sistema buscOpiniones

Este desarrollo[4] fue realizado como proyecto de grado por dos estudiantes, durante el año 2012.
Allí se implementa un sistema de recuperación de información sobre textos de prensa. Permite realizar búsquedas de opiniones, especificando una fuente y/o un asunto, a través de una interfaz web disponible en www.buscopiniones.com

Figure 3.1: Ejemplo de búsqueda de opiniones de Vázquez sobre Estados Unidos

Para hacerlo, utiliza el módulo identificador de opiniones presentado anteriormente. Además de la funcionalidad principal de búsqueda, permite elaborar reportes sobre los temas más mencionados en la prensa en un cierto momento del tiempo.

[2]Medida obtenida sin considerar los elementos reconocidos parcialmente

Los resultados obtenidos en las búsquedas son de **76%** de precisión, mientras que para la funcionalidad de los temas más mencionados obtienen un **86%** de precisión y un **84%** de recall.

Un problema que presenta este sistema es la identificación de fuentes. Actualmente el sistema busca literalmente la fuente que busca el usuario, sin tener en cuenta las opiniones emitidas por una fuente que no es la introducida por el usuario, pero sí es correferente con esta.

Este proyecto pretende brindar una solución a dicho problema obteniendo para cada artículo de prensa, un listado de fuentes de opinión, agrupado en clases de correferencias.

3.1.3 Una propuesta para la resolución de correferencias entre fuentes de opinión

Este trabajo[3] fue desarrollado como un proyecto de grado de tres estudiantes, en el año 2010.

El proyecto propone un módulo para la resolución de correferencias de fuentes para el idioma español. El trabajo parte de la salida del módulo identificador de opiniones y resuelve la recuperación de fuentes omitidas, así como las correferencias.

Algoritmo

La resolución se basa en un algoritmo de puntajes, definidos a partir de la información morfológica y sintáctica disponible en la salida del módulo identificador de opiniones.

En función de ciertas características como estar en una aposición, o tener un nombre propio o común con determinante definido o indefinido, las fuentes son elegidas como "primera mención" (iniciando una nueva cadena de correferencias), o como mención simple, buscando en este caso a cuál de las cadenas detectadas hasta el momento

pertenece.

Las características utilizadas en este trabajo son básicamente la comparación de lemas de núcleo,y la concordancia entre género, número y persona. También se utiliza Wordnet, aunque luego en los resultados se explica que utilizarlo decrementó la efectividad del programa.

Resultados

Los resultados son expresados utilizando la medida F tanto para la recuperación de fuentes, como para la resolución de correferencias[3].
El resultado se evalúa sobre textos de prensa, tomando la salida del módulo de opiniones intacta, así como corrigiéndola manualmente de forma de evaluar cuántos errores son del módulo y cuántos son de las herramientas previas que utiliza como entrada.

Para textos corregidos da resultados de *73.3%* de precisión y *61.1%* de recall para la recuperación de fuentes (medida F = **66.7%**), y *81.8%* en precision y *80%* en recall para la resolución de correferencias (medida F = **80.9%**).

Para textos corregidos, sin utilizar WordNet, da resultados de *84.6%* de precisión y *61.1%* de recall para la recuperación de fuentes (medida F = **70.9%**), y *81.0%* en precision y *85.6%* en recall para la resolución de correferencias (medida F = **83.2%**).

Para textos no corregidos[4] los resultados son de *40%* de precisión y *40%* de recall para recuperación de fuentes (medida F = **40%**), y *68%* de precisión y *66%* de recall para la resolución de correferencias (medida F = **67.4%**).

El proyecto presentado en este informe procurará mejorar este último resultado, priorizando el buen funcionamiento para textos no corregidos manualmente, ya que

[3]Esto define que la métrica utilizada para evaluar el proyecto presentado en este informe, sea también la medida F

[4]La entrada del módulo es la generada desde artículo de prensa, sin hacer ningún tipo de corrección

es imposible obtener la información de los programas previos sin ningún tipo de error.

3.2 Análisis del problema a resolver

El lugar donde deberá operar la solución brindada por este proyecto es el texto marcado por el módulo de opiniones, luego de procesar un artículo de prensa. A través del estudio de las opiniones encontradas y sus fuentes (identificadas o faltantes), se recuperarán las fuentes necesarias y se buscarán correferencias entre dichas fuentes.

La salida del módulo de opiniones consiste en un texto con marcas, donde se identifican:

- Los grupos nominales obtenidos por las reglas del módulo identificador de opiniones
- Opiniones (que pueden incluir a su vez, fuente, predicado, asunto o mensaje)

Además de las opiniones, el sistema puede realizar identificaciones parciales, identificando también los siguientes casos:

- Fuentes sin opinión
- Predicados sin opinión
- Predicados en una opinión que no contiene fuente

El objetivo de este proyecto es detectar y marcar las correferencias entre fuentes de opinión.

Esto implica en primer lugar lograr que a todos los predicados sin fuente asociada, se les asigne una. De este modo, es posible aumentar la cantidad de opiniones recuperables por el sistema buscOpiniones.
Llamamos a este problema **"Recuperación de Fuentes"**

Una vez resuelta esta tarea, procedemos a buscar correferencias entre todas las fuentes de opinión encontradas en el artículo.
Llamamos a este problema **"Resolución de Correferencias"**.

3.3 Recuperación de Fuentes

La recuperación de fuentes ocurre cuando tenemos un predicado que, o bien no está incluido en una opinión, o bien está incluido en una opinión que no tiene fuente definida.

A continuación se analizan varios casos y se muestra cómo se espera que este proyecto resuelva la fuente adecuada:

3.3.1 Características de las fuentes de opinión

Se espera que las fuentes candidatas sean elegidas de acuerdo con ciertas características que esperamos encontrar en una fuente de opinión. Estas son:

- Grupos nominales identificados como fuentes
- Grupos nominales que son aposiciones, presentando una nueva entidad en el discurso
- Grupos nominales que son sujeto en su oración
- Grupos nominales que contienen un nombre propio de persona u organización[5]
- Grupos nominales que contienen un nombre (propio o común)

Ejemplo:

*El presidente **Tabaré Vázquez** en conferencia de prensa dijo estar al tanto de la situación.*

[5]Notar que las características pueden coexistir en un candidato a fuente (ej: un nombre propio de persona,que además es sujeto de oración)

Dentro de la oración existen dos candidatos para el predicado "dijo". El más cercano es el nombre común "conferencia de prensa", que concuerda en número con "dijo", por lo cual no es un candidato a descartar. El siguiente candidato es "El presidente Tabaré Vázquez", que está a mayor distancia del predicado. El algoritmo deberá priorizar la aposición respecto al nombre común y, en general, priorizar los grupos nominales que sumen más características de fuente de opinión.

3.3.2 Concordancia

Deberá ocurrir que las fuentes elegidas coincidan en número con la conjugación del verbo elegido como predicado.

> ***Tabaré Vázquez**, en su exposición a **los diputados**, se **refirió** a los lineamientos presentados al asumir la presidencia, el primero de marzo.*

Podemos notar que el predicado "refirió" (singular) no concuerda con el grupo nominal candidato "los diputados" (plural), por lo que deberá ser descartado, pese a ser el candidato más cercano.

3.3.3 Citas Textuales

Deberá ocurrir que las fuentes elegidas se encuentren en el mismo nivel de cita que su predicado. Predicados que no están dentro de una cita textual no podrán ser vinculados a fuentes que sí están dentro de la cita textual.

> *"...donde fue convocado Daniel Martínez", agregó **Tabaré Vázquez**.*

En este caso, el grupo nominal "Daniel Martínez" no debe ser considerado como posible fuente, ya que no está en el nivel de alcance del predicado "agregó".

3.3.4 Distancia y Balance con las características del grupo nominal

Además de considerar las características propias de cada candidato, el algoritmo también deberá balancearlas con la distancia al predicado.

"El presidente de la república, Tabaré Vázquez, firmó el acuerdo el viernes pasado. El vocero de presidencia ***informó*** *que (...)*

En este caso, el predicado "informó" tiene varios grupos nominales candidatos: "El vocero de presidencia" (nombre común), "el viernes pasado" (nombre común), "el acuerdo" (nombre común), "El presidente de la república, Tabaré Vázquez" (Aposición).

En este caso, el nombre común "El vocero de la presidencia" deberá ser priorizado respecto a los otros candidatos, incluso sobre la aposición, debido a la distancia entre oraciones.

Otro punto a considerar es el lugar donde se buscan los candidatos. Además de buscar en todos los grupos nominales anteriores al predicado, deberá buscarse también entre el predicado y el punto final de su oración. De esta forma, se podrán resolver casos como el siguiente:

(...) ***informó*** [6] *el vocero de presidencia.*

3.3.5 Elipsis nominal

Es común encontrar elipsis nominales en artículos periodísticos, como forma de evitar la redundancia. Típicamente son oraciones donde encontramos al predicado de opinión en el comienzo de la oración.

"No es bueno que el Banco Central intervenga mucho en el mercado de valor del dólar", ***agregó****.*

[6] También deberá considerarse que si el predicado de opinión está seguido de la preposición "a", se debe descartar el grupo nominal que viene inmediatamente después.

En este caso, al no encontrar la fuente en la oración (ya que está omitida), el sistema deberá encontrar la fuente de opinión en las oraciones anteriores.

3.3.6 Conclusión

El desafío que se presenta para el algoritmo que resuelve este problema es balancear los criterios expuestos anteriormente para obtener buenos resultados en las diversas estructuras que se utilizan en los artículos periodísticos.

3.4 Resolución de Correferencias

En la sección 2.1 del capítulo anterior fueron analizados diversos tipos de referencias. A continuación se revisan ejemplos de correferencias que deberán ser resueltos por este proyecto:

3.4.1 Correferencia entre dos referencias idénticas

El ministro *expresó su (...).* ***El ministro*** *además agregó.*

Si las cadenas son exactamente iguales, se espera que las fuentes se marquen como correferentes.

3.4.2 Elementos anafóricos pronominales

Si bien los elementos anafóricos pronominales no son muy utilizados en el estilo periodístico, es posible que encontremos una referencia a través de un pronombre personal.

Ella *opinó que el Banco Central debería intervenir en el mercado de valor del dólar*

Debido al estilo de redacción empleado en artículos periodísticos, no se espera que este sistema resuelva anáforas pronominales. En caso de existir una anáfora pronominal, no será candidata a marcarse como fuente; el pronombre se saltea y se continúa la búsqueda de un grupo nominal que contenga como mínimo un nombre común.

3.4.3 Correferencia entre dos entidades reunidas en una aposición

También es un recurso muy común en dichos artículos a la hora de introducir nuevas entidades al discurso. En estos casos, típicamente se presenta un nombre propio acompañado de un grupo nominal.

El ministro de Transporte, **Victor Rossi**

Se espera que el sistema establezca una correferencia entre ambos miembros de la aposición, de forma que luego pueda ser referenciado tanto a través de "El ministro de transporte", como a través de "Victor Rossi".

3.4.4 Correferencia de tipo identidad, entre dos sinónimos

En estos casos, se utilizan dos formas distintas de referenciar a una misma entidad. El **ministro** de Transporte, Victor Rossi" luego es referenciado como "el **titular** de Transporte)

Este tipo de correferencia, para ser resuelta requiere tener fuentes de conocimiento del mundo.

3.4.5 Correferencia de tipo identidad, entre hipónimo e hiperónimo

En estos casos, para evitar la redundancia se utiliza una palabra más general para referenciar a la entidad ya mencionada. Típicamente es el hiperónimo el que se utiliza en la segunda referencia, y el hipónimo en la primera referencia.

El **ministro** de Transporte, Victor Rossi" luego es referenciado como "el **mandatario**" o "el **jerarca**

Este tipo de correferencia, para ser resuelta requiere tener fuentes de conocimiento del mundo.

3.4.6 Conclusión

Existen diversos casos de correferencias a resolver, y para ser correctamente resueltos, todos requieren de distintos tipos de información.

Respecto a la forma de medir el desempeño del algoritmo de resolución de correferencias, se profundiza sobre el tema en el Capítulo "Evaluación".

3.5 Solución propuesta (a nivel teórico)

La solución propuesta consiste en recuperar las fuentes a través de un algoritmo de puntajes similar a los descritos en la sección "Modelos de Puntajes". Cabe notar que la forma de resolver la fuente de un predicado de opinión (que es un verbo) resulta bastante análoga a la forma de buscar la fuente de un un elemento anafórico, ya que se tienen varios candidatos anteriores al predicado, se puede puntuar a cada candidato según ciertos atributos que esperamos encontrar en una fuente, y también se tienen restricciones entre los atributos del candidato y los del predicado de opinión.

Respecto a la resolución de correferencias, se utilizará un modelo de resolución entidad-mención, que emplea como características restrictivas, todas las definidas en la sección "uso de fuentes de conocimiento" : información gramatical, sintáctica y semántica, así como algunas relaciones entre cadenas de texto.

La estrategia de agrupamiento (*clustering*) utilizada es agrupar la fuente a la clase cuando resulta correferente con al menos un elemento de la clase (política "ANY"). Se utiliza dicha política debido a que los criterios para decidir correferencia son estrictos, hecho que se puede comprobar en la alta precisión lograda en los resultados.

Capítulo 4

Diseño de la solución

El diseño de la solución propuesta toma como base lo desarrollado en [8], donde los resultados del pre-procesamiento se almacenan en una base de datos relacional.

Por ello, el proyecto se divide en dos etapas independientes: una procesa y sintetiza la información de distintas aplicaciones de terceros en un modelo relacional, y la otra se vale de dicho modelo para resolver la tarea central de este proyecto, la resolución de correferencias.

4.1 Recuperación de fuentes y resolución de correferencias

Para esta etapa, podemos asumir que la información de las distintas aplicaciones ya se encuentra en la base de datos. Partiendo de esta base, se implementan los algoritmos necesarios para resolver las dos tareas presentadas en el capítulo anterior.

4.1.1 Recuperación de fuentes

Como se explicó anteriormente, es necesaria la recuperación de fuentes cuando identificamos una opinión que no tiene una fuente definida. Para ello, partimos de algunas premisas que limitan las fuentes candidatas dentro del texto:

- La fuente que buscamos siempre es un grupo nominal.
- La fuente que buscamos se encuentra o bien en la oración del predicado, o bien en una oración anterior.

En una primera instancia, todo grupo nominal que cumpla la segunda condición será candidato a ser fuente del predicado en cuestión.

El algoritmo propuesto evalúa uno a uno estos grupos nominales, define el subconjunto de los que son posibles por haber cumplido con ciertas condiciones, y finalmente, elige la fuente más probable a partir de la información extraída de otras aplicaciones.

Paso 1: Reducción del conjunto de candidatos

Para reducir el conjunto de posibles fuentes, se realizan dos validaciones:

Verificación de concordancia en número
Se evalúa el número del verbo elegido como predicado (singular, plural o indefinido) y el número del grupo nominal candidato. Se descarta únicamente cuando ambos tienen definido el número y éstos no coinciden. En otro caso, el grupo nominal sigue siendo candidato.

Se recuerda el ejemplo presentado en el capítulo anterior, cuando se habló de concordancia:

> ***Tabaré Vázquez****, en su exposición a* ***los diputados****, se* ***refirió*** *a los lineamientos presentados al asumir la presidencia, el primero de marzo.*

"Los diputados" es descartado debido a la no concordancia con "refirió".

Los nombres propios carecen de número en el sistema FreeLing. En este proyecto se decidió que todos los nombres propios fueran tratados como singular, debido a que es el uso más frecuente para nombres propios que son fuentes de opinión. Ejemplo:

*"**Estados Unidos** informó este jueves a los medios de prensa sobre (...)"*

Se puede notar que si bien el nombre propio claramente se trata de un plural (varios estados), en la gran mayoría de los casos va a ser referenciado como si fuera singular.
Verificación de zona válida
Se evalúa el contexto en el cual está el grupo nominal candidato, y se lo descarta si no cumple con alguna de las siguientes premisas:

- La fuente de un predicado (que no está en una cita) no puede buscarse dentro de citas textuales
- La fuente no puede estar incluída en texto que esté marcado como "asunto" por el módulo de detección de opiniones.

- La fuente no puede ser lo que sigue a un "a", "en", "que", si estas palabras están inmediatamente después del predicado.
  Ejemplo: "**dijo** a El Observador la ministra.", en este caso se descarta "El Observador" como candidato.

Paso 2: Elegir la fuente candidata

Una vez que logramos reducir el conjunto de posibles fuentes del predicado, debemos elegir uno de ellos. Para ello, tenemos en cuenta las siguientes características, agrupadas en dos conjuntos:

Se analiza la distancia del grupo nominal candidato al predicado:

- Distancia al predicado (en palabras).
- Distancia al predicado (en oraciones).

Se analiza la naturaleza del grupo nominal, evaluando lo siguiente:

- Si el grupo nominal está marcado como fuente
- Si se trata de una aposición
- Si contiene un nombre propio de persona u organización
- Si contiene un nombre propio
- Si contiene un nombre común
- Si el grupo nominal es o contiene al sujeto de la oración

La función que puntúa cada candidato deberá balancear adecuadamente estos dos conjuntos ya que, a modo de ejemplo, un nombre propio a una distancia muy corta del predicado, puede ser tan relevante como una aposición ocurrida en la oración anterior.

Para encontrar las ponderaciones adecuadas, se utilizó la planilla de cálculos mostrada en la siguiente imagen:

		0	20	40	100	25
	Puntaje	Muy anterior	Dos oraciones antes	Una oración antes	En la oración	En zona +-5 del predicado
FUENTE	150	150	170	190	250	275
GN APOSICION	140	140	160	180	240	265
GN SUJETO NP PERSONA	100	100	120	140	200	225
GN SUJETO NP	90	90	110	130	190	215
GN NP PERSONA	90	90	110	130	190	215
GN SUJETO	70	70	90	110	170	195
GN NP	50	50	70	90	150	175
GN	20	20	40	60	120	145
	En verde se muestran los casos que serían priorizados respecto a un candidato con el puntaje ingresado				200	

Figure 4.1: Ejemplo de consulta en la tabla de balance de puntajes

Los valores iniciales fueron definidos sobre un conjunto de prueba de cinco artículos de prensa.

En este ejemplo se consulta cuáles entidades y a qué distancia estarían superando a un grupo nominal sujeto de oración que contiene nombre propio de persona. Quedan resaltadas en verde las combinaciones que podrían ser priorizadas respecto a dicha clase de grupo nominal.

Esto permitió calibrar algunos puntajes de forma que hubiera cierto equilibrio entre la naturaleza del grupo nominal y la distancia. Para cada tipo de grupo nominal, se verificó que los candidatos que se priorizarían realmente fueran mejores candidatos.

Una vez realizada dicha verificación, se probó nuevamente sobre quince nuevos artículos de prensa donde se realizaron algunos ajustes menores y se logró una calibración estable.

Logrado esto, se implementa esta función que puntúa cada grupo nominal candidato en función de las características anteriores, y se escoge aquél candidato que maximice dicha función.

A continuación se presenta un pseudocódigo de la función mencionada:

Algoritmo 1 Puntuación de un Grupo Nominal candidato

```
function PUNTUAR_FUENTE(GN)

    if [GN está marcado como fuente de una opinión] then
        Puntaje ← 150
    else if [GN se reconoce como una aposición] then
        Puntaje ← 140
    else if [GN contiene un NP de pers/org, que es sujeto de la oración] then
        Puntaje ← 100
    else if [GN contiene un NP, que es sujeto de la oración] then
        Puntaje ← 90
    else if [GN contiene un NP de pers/org] then
        Puntaje ← 90
    else if [GN contiene al sujeto de la oración] then
        Puntaje ← 70
    else if [GN contiene un NP] then
        Puntaje ← 50
    else
        Puntaje ← 20
    end if

    if [GN en la misma oración que el predicado] then
        Puntaje ← Puntaje + 100
        if [GN está a menos de 5 palabras del predicado] then
            Puntaje ← Puntaje + 25
        end if
    else if [GN en oración anterior al predicado] then
        Puntaje ← Puntaje + 40
    else if [GN dos oraciones atrás del predicado] then
        Puntaje ← Puntaje + 20
    else if [GN más de cinco oraciones atrás del predicado] then
        Puntaje ← Puntaje − 20
    end if
end function
```

4.1.2 Resolución de correferencias

Una vez culminado el proceso de recuperación de fuentes, se cumple que cada predicado de opinión tiene una fuente asociada. La segunda tarea a resolver, consiste en determinar cuándo estas fuentes son correferentes.

Para ello, deben resolverse dos aspectos importantes:

- Definir cuándo dos fuentes de opinión son correferentes
- Definir una política de agrupamiento: resolver cuándo una fuente de opinión es parte de una clase (compuesta de varias fuentes correferentes) y cuándo no lo es.

Correferencia entre dos fuentes

Como se comentó en la solución propuesta en el capítulo anterior, la resolución de correferencias estará basada en distintas fuentes de conocimiento: información gramatical, sintáctica, semántica, etc.
Los criterios utilizados para decidir si dos fuentes correfieren son los siguientes:

Identidad

Decimos que si dos fuentes de opinión son exactamente la misma, se refieren a la misma entidad del mundo.

Inclusión

Si la cadena de texto de una de las fuentes está enteramente incluida en otra fuente, decimos que existe correferencia.
Ejemplo: "Mujica" será correferente con "El senador Mujica", debido a inclusión completa.

Distancia de Levenshtein

La distancia de Levenshtein se utiliza para medir entre dos palabras, cuál es la cantidad mínima de inserciones/borrados/modificaciones que tengo que hacer para transformar una en la otra.

Ejemplo: La distancia de Levenshtein entre "HOLA" y "KOALA" es dos, ya que puedo insertar la A entre la O y la L ("HOALA") y luego sustituir la H por una K ("KOALA").

Debido a los posibles errores de escritura (por ejemplo al escribir nombres foráneos), se utiliza una distancia menor o igual a dos como umbral para decidir si se trata de una misma palabra mal escrita, y por tanto asumir correferencia.

Filtro de nombres propios distintos

Si dos fuentes de opinión contienen nombres propios de persona u organización, y no tienen en común a ninguno de ellos, se decide que las dos fuentes NO son correferentes. Ejemplo: "El ministro Danilo Astori" y "El ministro Victor Rossi", si bien comparten el nombre común "ministro", se descarta correferencia por contener distintos nombres propios de persona u organización.

Compartir nombre propio de persona u organización

En casos que el filtrado anterior se supere, cuando existe algún nombre propio de persona u organización en común, nos basta para decidir que las fuentes sí son correferentes.

Ejemplo: "El ex-presidente José Mujica" y "El senador José Mujica", se marcan como correferentes por compartir el nombre propio de persona "José Mujica"

Relaciones semánticas entre los nombres

Para buscar relaciones semánticas entre dos fuentes de opinión, primero se extraen todos los nombres contenidos en cada opinión (sean nombres propios o nombres comunes). Luego, para cada par de nombres, se busca si existe algún tipo de relación léxica o semántica entre ellas (por ejemplo, sinonimia, hiperonimia, etc.).

No se presentará pseudocódigo de esta función, ya que su implementación no guarda aspectos de interés que no se hayan mencionado en el listado anterior.

Política de agrupamiento de fuentes correferentes

Una vez que se tienen implementados los criterios anteriores, es necesario definir de qué forma se van a agrupar las clases de correferencias.

Considerando que este proyecto tiene criterios exigentes para afirmar correferencia (es un algoritmo que sin dudas brinda más precisión que recall), cuando hay dos fuentes marcadas como correferentes, la probabilidad de que lo sean es alta.

Por lo tanto, se define que para que una fuente f_k pertenezca a la clase $[f_1, f_2, ..., f_n]$, basta con que exista una correferencia entre f_k y f_i, para algún i entre 1 y n.
Dicha política es la definida en el marco teórico como "ANY".

Capítulo 5

Implementación

La primera etapa de la implementación del sistema descrito en este informe consistió en resolver la integración de información proveniente de distintas aplicaciones.

Para ello, toma como entrada un artículo de prensa escrito en un archivo de texto plano y lo analiza utilizando distintas aplicaciones externas, obtienendo de cada una de ellas un conjunto de atributos a nivel de palabra, conjunto de palabras u oración.

Finalmente se implementa un módulo que resuelve las correferencias entre fuentes de opinión, valiéndose de la información obtenida en la etapa de pre-procesamiento.

En la primera etapa, la integración fue realizada como indica el siguiente diagrama:

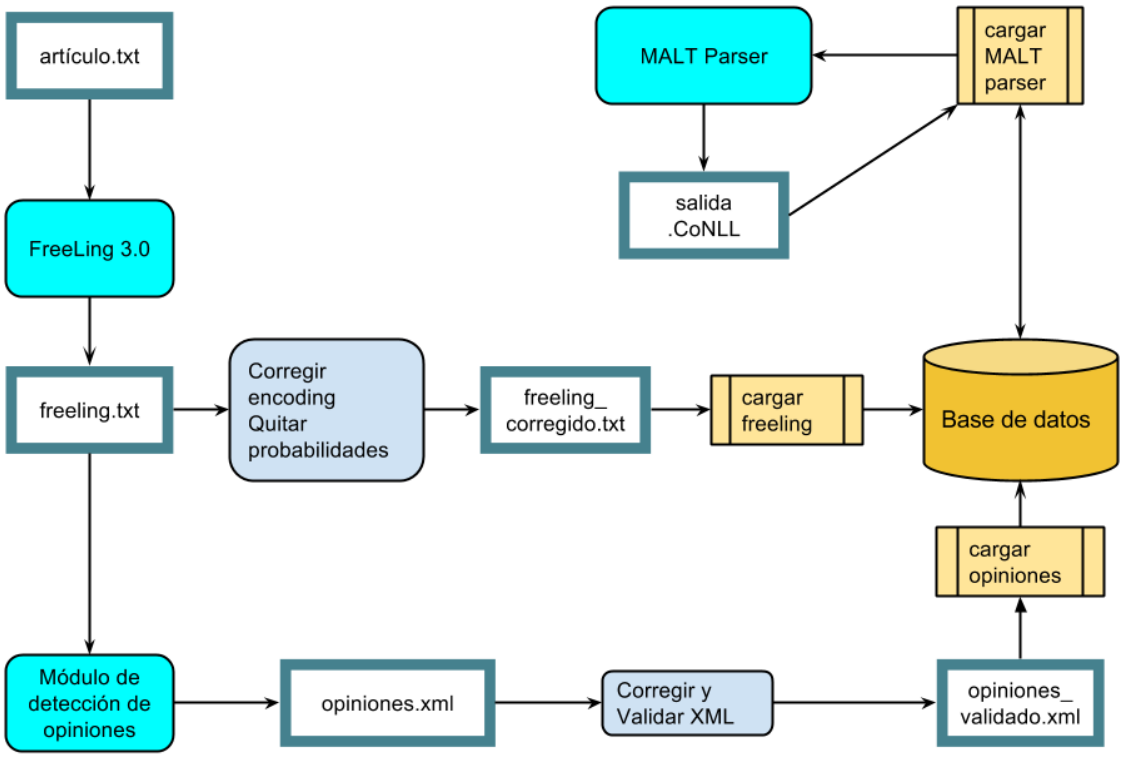

Figure 5.1: Diagrama de la integración realizada

Las salidas obtenidas de estas aplicaciones son procesadas, e integradas a una base de datos similar a la definida en[8].

5.1 Modelo de la base de datos

La base de datos se compone de tres tablas utilizadas para almacenar atributos obtenidos de otras aplicaciones, y una tabla para almacenar los resultados de la anotación de fuentes y de la resolución de correferencias.
A continuación se presenta el modelo de base de datos:

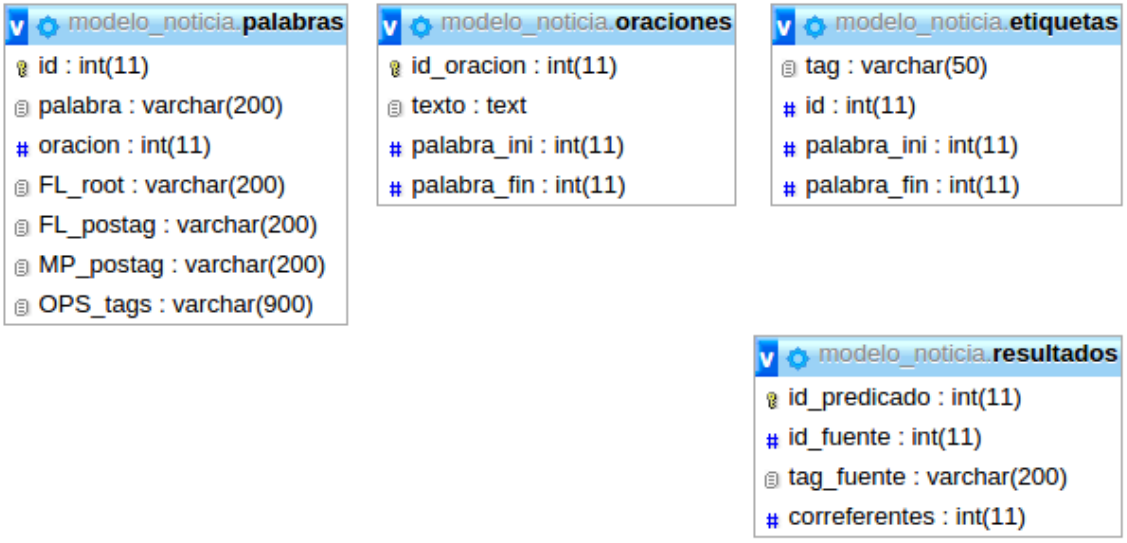

Figure 5.2: Modelo de la base de datos relacional

5.1.1 Tabla de Palabras

La tabla de palabras se completa a partir de la salida obtenida de FreeLing. Los índices definidos por FreeLing son incrementales e identifican de forma unívoca a cada palabra en el texto.
La función de esta tabla es almacenar aquellos atributos que aplican individualmente a las palabras encontradas en el texto.

5.1.2 Tabla de Oraciones

La función de la tabla de oraciones es almacenar las distintas oraciones detectadas en el texto original. Además del identificador, se registra el identificador de su primera palabra y de su última palabra.
Es utilizada para invocar al parser de dependencias.

5.1.3 Tabla de Etiquetas

La tabla de etiquetas se utiliza para almacenar todas aquellas marcas que abarcan varias palabras del texto (ej: fuentes, grupos nominales, predicados, opiniones).
Al procesar la salida de las distintas aplicaciones externas, se van agregando las etiquetas a esta tabla.

5.2 Análisis utilizando FreeLing 3.0

5.2.1 Sobre la aplicación

FreeLing 3.0 es una librería de código abierto para el análisis de textos multi-lenguaje. Además de algunas funcionalidades básicas como la tokenización y la partición en párrafos y oraciones, permite realizar con buena efectividad tareas como: lematización, análisis morfológico y etiquetado gramatical de cada palabra (a través del PoS tag), así como de detección y clasificación de nombres de entidades.

También cuenta con algunas funcionalidades experimentales para parseo de dependencias, parseo superficial (*shallow parsing*) y resolución de correferencias, que reportan medidas sub-óptimas respecto al estado del arte de dichos problemas, ya que aún se encuentran en desarrollo (ver[12]).

En este proyecto, se utiliza el análisis morfológico y el etiquetado gramatical.
La versión 3.0 de FreeLing aporta una categorización de los nombres propios, clasificándolos en "Nombre Propio", "Organización", "Lugar geográfico", etc. Más adelante se explicará cómo esta nueva categorización afecta positivamente la recuperación de fuentes.

5.2.2 Motivación

FreeLing será de utilidad para las siguientes tareas:

- Para obtener las categorías gramaticales de cada palabra y encontrar los nombres y nombres propios.
- Para utilizar la información de género y número, y entonces verificar concordancias.
- Para ponderar mejor los nombres propios de persona u organización.

5.2.3 Integración al proyecto

Al analizar el texto original, obtenemos el resultado en un archivo plano, organizado en forma de tabla. A cada palabra corresponde una línea del archivo, donde se encuentran las características definidas por FreeLing, separadas por tabuladores.

A modo de ejemplo, si la palabra "opinó" es parte del texto, tendría una línea en el archivo de salida, y contendría las siguientes características:

Atributo	**Explicación**
opinó	Palabra original.
opinar	Raíz de la palabra.
VMIS3S0	POS-tag asignado, ver convención de EAGLES[27].
1	Probabilidad de que el POS-tag sea el adecuado (puede no ser 1 cuando existe ambigüedad en la palabra).

Esta es toda la información que nos brinda FreeLing sobre esa palabra en esa oración del texto. Para ingresar la información a la base de datos, basta con procesar el archivo de salida de FreeLing y actualizar los campos *FL_root* y *FL_postag* de cada palabra, en la tabla de palabras.

5.3 Módulo de detección de Opiniones

5.3.1 Sobre la aplicación

El módulo que se encarga de identificar las opiniones que ocurren en el texto está implementado en Prolog[30], y toma como entrada la salida de FreeLing.
El módulo utiliza un enfoque por reglas, incluyendo un lexicón para los predicados[7].

Las opiniones se componen de los siguientes elementos:

Fuente

De ser posible, identifica la entidad que emitió la opinión detectada. De lo contrario, se agrega una fuente vacía que será resuelta por el módulo de correferencias. Gramaticalmente, en la mayoría de los casos se trata del sujeto de la oración que contiene la opinión.

Predicado

El predicado identifica la expresión (típicamente un verbo) que transmite la opinión. Posibles predicados de opinión son: "dijo", "afirmó", "opinó", "indicó", "se refirió","añadió".

Mensaje

El mensaje identifica a la opinión en cuestión. Puede contener por ejemplo, texto citado entre comillas o una frase subordinada introducida con una conjunción subordinante como "que".

Asunto

El asunto es un complemento del predicado que busca identificar el tema sobre el cual se está opinando.

No es necesario identificar los cuatro componentes para que una opinión sea detectada; es posible que una opinión contenga, por ejemplo, un predicado y un mensaje, pero la fuente no esté definida.

La salida está en formato xml. Ejemplos de opiniones detectadas:

```
<opinion>
<fuente>El candidato a vicepresidente</fuente>
<predicado>dijo</predicado>
<mensaje>que realiza un "balance muy positivo" de este viaje</mensaje>
</opinion>
```

Figure 5.3: Opinión identificada, con fuente y mensaje. Asunto no encontrado.

```
<opinion>
<fuente> </fuente>
<predicado>destacó</predicado>
<asunto>la intención de seguir manteniendo una buena relación con el FMI</asunto>
</opinion>
```

Figure 5.4: Opinión identificada, con asunto. Fuente no encontrada.

5.3.2 Motivación

Debido a que el objetivo principal de este proyecto es resolver las correferencias entre las fuentes de las opiniones de un texto en español, la información provista por este módulo es el eje central de este proyecto.

5.3.3 Integración al proyecto

Lo primero que se realizó para integrar el módulo de opiniones al proyecto, fue adaptar la salida del nuevo FreeLing a la salida esperada, quitando la columna de probabilidad.

Segundo, la interacción con el módulo de opiniones es principalmente a través de consola, donde el usuario elige el archivo de entrada y el archivo de salida. Se desarrolló un script que permite utilizar el módulo de forma automática, ya que uno de los objetivos de este proyecto es integrar todos los módulos involucrados para que el procesamiento del texto en ningún momento requiera intervención de usuario.

Tercero, la información provista por este módulo aporta información a nivel de palabra, además de ingresar las etiquetas en la tabla correspondiente:

Tabla de palabras

En esta tabla se guardan todas las marcas bajo las cuales está etiquetada cada palabra. Por ejemplo, si en el texto tenemos la fuente "El presidente", es posible que esté marcada como un **gn** (grupo nominal), también que sea parte de una fuente, y que sea parte de una opinión. Se guardará un string con todas las marcas asignadas por el módulo.

Tabla de etiquetas

Cada etiqueta de opinión, predicado, asunto, o fuente, es agregada a la tabla de etiquetas. De esta forma, luego será posible iterar sobre ellas a la hora de recuperar las fuentes.

5.4 Análisis utilizando el MALT Parser

5.4.1 Sobre la aplicación

Para realizar un análisis sintáctico de la estructura de una oración, disponemos de distintos tipos de parsers. La tarea de un parser consiste en evaluar la estructura de una oración respecto a una gramática, y su tipo varía según el tipo de la gramática que utiliza.

En este caso se utilizará un parser de dependencias. Su gramática agrupa las palabras entre sí según las dependencias encontradas. El resultado es un árbol formado exclusivamente por las palabras de la oración, teniendo como raíz la palabra etiquetada como *ROOT* y luego cada nodo agrupa como hijas a aquellas palabras que tenga como subordinadas.

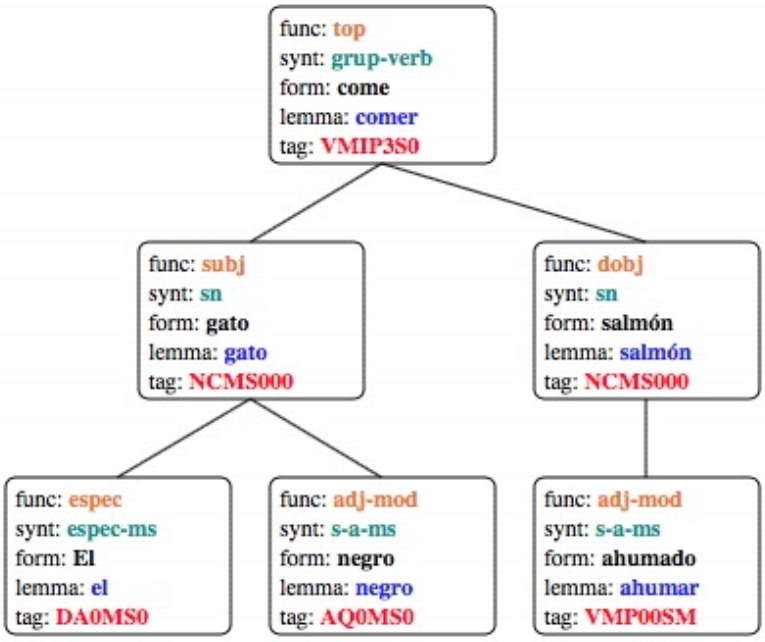

Figure 5.5: Oración analizada con un parser de dependencias

El MALT parser [13] es una herramienta que utiliza técnicas de aprendizaje automático para generar distintos modelos de parseo a partir de un corpus de datos

(de un *treebank*). Originalmente fue desarrollado para el idioma inglés, por lo que los modelos pre-entrenados que trae por defecto, están disponibles para el inglés y algunas otras lenguas, entre las cuales no se encuentra el español.

El Instituto Universitario de Lingüística Aplicada (IULA) de la Universitad Pompeu Fabra desarrolló un corpus de 41.232 oraciones en español anotadas con información sintáctica[14], que fue utilizado para crear un parser de dependencias para el español (espmalt.mco).

Este proyecto utiliza dicho modelo para obtener el árbol de dependencias para cada oración del texto procesado.

5.4.2 Motivación

El MALT parser brindará información complementaria respecto a la función que tiene en su oración cada nombre candidato a ser fuente. Nos interesa particularmente por dos razones:

- Para expandir los grupos nominales encontrados por el módulo de opiniones
- Para ponderar aquellos nombres propios que sean sujetos de la oración, respecto a los que sean nombres propios pero tengan una función más complementaria en la oración.

5.4.3 Integración al proyecto

Para integrar el MALT parser, se analizaron dos alternativas: consumir desde la aplicación el servicio web provisto por IULA[28], o descargar el software y utilizarlo en la misma máquina donde se ejecuta este proyecto.

Debido a que el servicio web tiene un límite en los tamaños de texto que procesa, por una cuestión de enlentecer lo menos posible la etapa de pre-procesamiento, se optó por descargar la versión y procesar el archivo .CoNLL generado por el MALT

parser.

Dicho formato simplemente es un texto plano separado por tabuladores, similar a la salida de FreeLing:

```
1       El      el      d       DA0MS0  _       2       SPEC    _       _
2       gato    gato    n       NCMS000 _       4       SUBJ    _       _
3       negro   negro   a       AQ0MS0  _       2       MOD     _       _
4       come    comer   v       VMIP3S0 _       0       ROOT    _       _
5       salmón  salmón  n       NCMS000 _       4       DO      _       _
6       ahumado ahumar  v       VMP00SM _       5       MOD     _       _
7       .       .       f       Fp      _       6       punct   _       _
```

Figure 5.6: Ejemplo de salida en formato CoNLL

Se puede ver que utilizando las referencias de la séptima columna es posible armar el árbol de dependencias de la figura anterior.

Para cargar esta información a la base de datos, se procede de forma análoga al procesamiento del archivo de salida de FreeLing: se unifica la fila asociada a la palabra y se guarda el string en el campo *MP_postag*. De esta forma, en la siguiente etapa podremos saber qué función tiene una palabra en su oración, así como sus dependencias en ambos sentidos, según el arbol de dependencias.

Vale aclarar que actualmente los parsers de dependencia no brindan buenos porcentajes de acierto al armar el árbol de dependencia, típicamente el árbol queda parcialmente correcto aunque con algunos errores. De todas formas este proyecto los utiliza únicamente para tareas muy concretas (completar grupos nominales, y detectar sujeto de la oración).

5.5 Análisis del texto utilizando Wordnet 3.0 en español

5.5.1 Sobre la aplicación

Wordnet[15] surge como un proyecto del Departamento de Ciencias Cognitivas de la Universidad de Princeton, que se propone crear una base de datos de palabras del idioma inglés y almacenar sus agrupamientos y relaciones de forma similar a como lo hace la memoria humana.

Los sustantivos, verbos y adjetivos se organizan en conjuntos de sinónimos denominados *synsets*. Cada synset busca representar un concepto, que puede ser referenciado por cualquiera de sus palabras.

A su vez, también se registran las relaciones entre estos conjuntos, para representar por ejemplo hipónimos/hiperónimos, holónimos/hiperónimos, etc.

Con el objetivo de ampliar Wordnet a distintos idiomas, surge el proyecto *Multilingual Central Repository*[16]. Este proyecto no solamente busca agregar léxico de otras lenguas, sino que también busca vincular las palabras de un idioma a aquellas de otros idiomas con las que comparta significado.

La estructura de las tablas de Wordnet para español son las siguientes:

Tabla de synsets
: Contiene el conjunto de synsets y su significado. Cada synset está identificado por un offset (o desplazamiento) en el archivo diccionario de Wordnet.

Tabla de variantes
: Contiene todas las palabras registradas, y en cuáles synsets está incluida.

Tabla de relaciones
: Contiene las relaciones semánticas entre dos synsets. Para cada par de synsets relacionados, provee el tipo de relación.

5.5.2 Motivación

La información de Wordnet ayudará al resolver correferencias. Se podrán vincular dos fuentes que lingüísticamente no comparten lema ni otra información (ejemplo: "presidente" y "mandatario") pero sí están vinculadas semánticamente.

5.5.3 Integración al proyecto

Para integrar Wordnet al proyecto manteniendo la idea de esta etapa de pre-procesamiento (extraer a la base de datos la información relacionada al artículo), en un principio se optó por guardar los synsets a los cuales pertenece cada palabra del artículo en una tabla, y luego buscar las relaciones entre los synsets de cada par de palabras del artículo, y guardarlas en otra tabla.

Tras evaluar el costo en tiempo de procesamiento y el uso de Wordnet que efectivamente hace el módulo de correferencias, se decidió que este procesamiento de WordNet, por defecto estuviera deshabilitado, y que en su lugar existieran funciones que permitan hacer la consulta utilizando las tablas de su base de datos.

Sin embargo, el procesamiento de Wordnet en esta etapa puede habilitarse si se desea utilizar este módulo para desarrollar sobre el un proyecto de otra naturaleza, que requiera un uso más intensivo de WordNet.

5.6 Conclusiones de esta etapa

5.6.1 Resumen

Al cierre de esta etapa de pre-procesamiento del artículo, disponemos de una base de datos con toda la información lingüística provista por FreeLing, el MALT parser, el módulo de Opiniones y Wordnet.

Para modularizar esta etapa, y facilitar la tarea de quien quiera utilizar la información cargada en la etapa de pre-procesamiento, se implementó un conjunto de funciones que resuelve ciertas consultas de interés para el desarrollador.

Esto genera un avance importante para el desarrollo de nuevos módulos que quieran utilizar la información de FreeLing, MALT parser y del módulo de opiniones.

5.6.2 API de consultas a la información de la base de datos

Esta API permite obtener, por ejemplo, una única estructura de datos con el texto completo y la información de sus palabras, o todos los grupos nominales detectados en una cierta sección del texto, sin tener que conocer el lenguaje de consultas del servidor de base de datos que se está utilizando.

Además de las funciones de consulta básica sobre tablas, se agregaron varias funciones definidas para resolver algunas tareas sencillas relacionadas con el procesamiento. Para mayor detalle, ver **Apéndice A**

Una vez implementado el pre-procesamiento del artículo, se implementa el módulo de resolución de correferencias haciendo uso de las funciones de la API.

A continuación se presentan pseudocódigos que muestran la estructura básica de la implementación, tanto para la recuperación de fuentes como para la resolución de correferencias.

5.7 Algoritmo para la Recuperación de Fuentes

Notar que primero se eligen candidatos entre los grupos nominales que fueron marcados como fuente por el identificador de opiniones, y luego, se eligen en el resto de los grupos nominales encontrados en el texto.

Algoritmo 2 Recuperación de fuentes

```
Candidatos ← ∅
Predicados ← OBTENER_PREDICADOS_NO_RESUELTOS( )

for each P in Predicados do
    Alcance ← FIN_ORACION_DE_PREDICADO(P)

    Fuentes ← OBTENER_FUENTES(0, Alcance)
    for each F in Fuentes do
        if (verifica concordancia) and (verifica zona válida) then
            Candidatos ← F
        end if
    end for

    GruposN ← OBTENER_GRUPOS_NOMINALES(0, Alcance)
    for each G in GruposN do
        if (verifica concordancia) and (verifica zona válida) then
            Candidatos ← G
        end if
    end for

    FuenteElegida ← SELECCIONAR_MEJOR_FUENTE(Candidatos)
    RESOLVER_PREDICADO(P,FuenteElegida)
end for
```

La función **Seleccionar_Mejor_Fuente** simplemente escoge de entre los grupos nominales candidatos, a aquel que tenga mayor puntaje según la función **Puntuar_Fuente**.

5.8 Algoritmo para el agrupamiento de fuentes correferentes

Notar que se utiliza la política "ANY" descrita en el capítulo anterior:

Algoritmo 3 Resolución de Correferencias

$[f_1, f_2, ..., f_n] \leftarrow$ OBTENER_FUENTES_DE_OPINION( )

for $i = 0$ to n **do**
  $fuente\_agrupada \leftarrow FALSE$
  for $j = 0$ to $i - 1$ **do**
    if SON_CORREFERENTES(f_i,f_j) **then**
      $clase\_de\_j \leftarrow$ OBTENER_CLASE(f_j)
      AGREGAR_FUENTE_A_CLASE(f_i,$clase\_de\_j$)
      $fuente\_agrupada \leftarrow TRUE$
    end if
  end for
  if ($fuente\_agrupada == FALSE$) **then**
    INICIAR_NUEVA_CLASE(f_i)
  end if
end for

Capítulo 6

Evaluación del sistema

En este capítulo se presenta la evaluación del sistema implementado. Los resultados reportados por el sistema serán contrastados con una corrección manual de los artículos.

Para esto, se analizará en primer lugar el desempeño en la tarea de recuperación de fuentes, utilizando como métrica la medida F.

En segundo lugar, se repasarán las distintas métricas utilizadas para evaluar sistemas de resolución de correferencias, para luego analizar el desempeño de este proyecto.

La evaluación presentada en este capítulo fue realizada sobre 45 textos de prensa elegidos al azar, de diversos medios de prensa escrita uruguaya (diario El Observador, diario El Pais, portal Montevideo Comm). En total se recuperaron 184 fuentes sobre un total de 188 opiniones sin fuente, y se resolvieron 84 correferencias, de un total de 101 correferencias.

6.1 Método de evaluación de la recuperación de fuentes

El objetivo de esta sección consiste en evaluar qué tan bueno es el sistema al identificar la fuente de los predicados que no tienen una asignada.

Para ello, vamos a suponer que tenemos **n** predicados a resolver: [p_1, p_2, ..., p_n].

Sean [f_1, f_2, ..., f_n] las respectivas fuentes recuperadas por el sistema, y [F_1, F_2, ..., F_n] las fuentes correctas (encontradas por una persona, verificando el texto).

Cabe destacar que tanto f_i como F_i pueden ser vacías, ya que es posible que el sistema no encuentre ningún grupo nominal viable, así como es posible que el predicado no tenga una fuente asignada (predicados que si bien surgen de un verbo contenido en el lexicón de predicados, carecen de fuente. Ej: "En caso de **confirmar** lo anterior...").

El método de evaluación empleado es a través de la medida F, obtenida a partir de la precisión y el recall del sistema.

Medida F

Esta métrica surge de combinar a través de la media armónica, los valores de precisión y recall del sistema. Estos valores son números entre 0 y 1, y miden distintas características del sistema.

Precisión

La precisión busca medir cuántas veces el sistema identificó correctamente una fuente, tomando como base la cantidad de identificaciones realizadas.

Podemos escribir la precisión obtenida como:

$$\mathbf{P} = \frac{|\{f_i | f_i \text{ coincide con } F_i\}|}{|\{f_i\}|}$$

Recall

Complementariamente, el recall busca medir cuántas veces el sistema identificó correctamente una fuente, tomando como base la cantidad de predicados con resolución[1] que contiene el sistema.

Podemos escribirlo de la siguiente manera:

$$\mathbf{R} = \frac{|\{f_i | f_i \text{ coincide con } F_i\}|}{|\{F_i\}|}$$

Es deseable que el sistema posea buenas medidas en ambas características, dado que una buena precisión acompañada de un mal recall nos da un sistema que deja sin resolver demasiados predicados, pese a hacerlo muy bien las pocas veces que lo hace. La mayoría de los errores son de tipo "falso negativo".

Si tenemos un buen recall pero baja precisión, vamos a tener un sistema que resuelve todos los casos que debería resolver, a costa de resolver erróneamente casos que no debería resolver. La mayoría de los errores son de tipo "falso positivo".

Para equilibrar estas dos medidas, se utiliza la medida F_1, que se expresa de la siguiente manera:

$$\mathbf{F} = \frac{2 * P * R}{P + R}$$

[1] Los predicados identificados de forma errónea (por ejemplo, nombres como "consideración", del verbo considerar, que es predicado de opinión) no se consideran ni correctos ni erróneos al evaluar la recuperación de fuentes

6.2 Resultados de la etapa de Recuperación de Fuentes

En esta sección se exponen y analizan los resultados obtenidos en la recuperación de fuentes.

Resultados

A continuación se presenta la cantidad de predicados que requirieron recuperación de fuentes durante la evaluación:

Cantidad de predicados sin fuente	**Cantidad de predicados resueltos**	**Cantidad de predicados resueltos correctamente**
188	184	138

Obteniendo los siguientes resultados para la medida F:

Precisión	**Recall**	**Medida F**
75%	73%	74%

Análisis

En principio, el resultado obtenido para la recuperación de fuentes es bueno. Se recuerda que el mejor resultado obtenido por [3] fue de F=70.95%, obtenido al corregir manualmente las salidas de las aplicaciones involucradas (FreeLing, Módulo de opiniones).

Esto puede deberse a varios factores:

- Utilización de un parser de dependencias para ponderar mejor los sujetos de oración.
- Mayor cantidad de grupos nominales candidatos, ya que también se obtienen los identificados por el parser de dependencias.
- Contar con la identificación de nombres propios de persona u organización por parte de FreeLing 3.0

Errores encontrados, y sus posibles causas

El sistema evaluó la recuperación estricta de las fuentes, por lo que en algunas ocasiones (tres) la recuperación de la fuente fue parcial, lo cual se consideró incorrecto. Ejemplo:

El Pentágono difundió la transcripción de las declaraciones efectuadas por ***el presunto*** *líder terrorista , quien* ***confesó*** *su responsabilidad (...)*

En dicho ejemplo, "el presunto" fue marcado como fuente candidata en lugar de "el presunto líder terrorista", ya que no fue correctamente expandido a partir del árbol de dependencias.

También aparecieron algunos casos donde el filtro de zona válida no fue lo suficientemente flexible para prevenir ciertos errores. Ejemplo:

"Estoy sorprendido por el tono empleado por el Dr. Larrañaga y que haya elegido hacer su planteo por la prensa", ***dijo*** *ayer en* ***el Consejo de Ministros****.*

En este caso, la palabra "ayer" hace que no se utilice la regla de descartar lo que sigue a la preposición "en", contigua al predicado. Por lo tanto, se eligió como fuente "el Consejo de Ministros"
Otro error similar que podría corregirse afinando las zónas válidas:

A los casi 31 años , Natalia Oreiro lleva dos tercios de su vida de exposición mediática . Mientras conduce un programa ecologista y espera cerrar un contrato para un filme de un director extranjero , ***habló*** *con* ***la revista Para Ti*** *acerca de (...)*

Al tratarse de una elipsis, ya que la fuente original (Natalia Oreiro) no se encuentra en la oración, el grupo nominal "la revista Para Ti", que contiene un nombre propio cobra mayor relevancia y se elige como fuente del predicado "habló".

Otro tipo de error surge de los errores de ponderación, en particular de ponderar demasiado los grupos nominales que fueron marcados como fuentes de opinión. Ejemplo:

Un juez federal de la provincia de Mendoza ***confirmó*** *la verdadera identidad de Celina Manrique luego de que un miembro de su familia adoptiva iniciara una investigación ante las sospechas que le despertó la telenovela , indicó* ***el diario Clarín de Buenos Aires****.*

El predicado "indicó" está correctamente resuelto, ya que la fuente es el diario mencionado . Pero al resolver el predicado "confirmó", el grupo nominal "Un juez federal", quien es sujeto de la oración aunque no contiene nombre propio queda en segundo lugar, ya que el diario clarín está marcado como fuente de opinión.

Conclusión respecto a los errores encontrados

En términos generales se puede aspirar a evitar varios de los errores encontrados, de dos formas: en primer lugar afinando la función que define las zonas válidas para un predicado, y en segundo lugar mejorando la calibración de los puntajes.

6.3 Métodos de evaluación de sistemas que resuelven correferencias

Existen diversas métricas para evaluar sistemas de resolución de correferencias en textos, por lo que se hará un breve repaso sobre algunas de ellas en base a la información presentada en [18] y [19].

6.3.1 Medida F

Esta métrica es la utilizada en [3], y se calcula de la siguiente forma:

Si el sistema detectó N pares de fuentes correferentes:

$$S = \{s_0, s_1, ..., s_N \text{ donde } s = (f_i, f_j) \textbf{ par posiblemente correferente}\}$$

Y si corrigiendo el texto manualmente, se detectaron M pares de fuentes correferentes:

$$K = \{k_0, k_1, ..., k_M \text{ donde } k = (f_i, f_j) \textbf{ par correferente confirmado}\}$$

calculamos la precisión y el recall del algoritmo como:

$$\mathbf{P} = \frac{|S \cap K|}{|S|} \quad \mathbf{R} = \frac{|S \cap K|}{|K|}$$

Y utilizamos nuevamente la medida F definida como:

$$\mathbf{F} = \frac{2 * \mathbf{P} * \mathbf{R}}{\mathbf{P} + \mathbf{R}}$$

6.3.2 Métrica $MUC-6$

Esta métrica también calcula precisión y recall globales, pero utilizando otro enfoque: se miden cuántos inserciones y borrados son necesarios para obtener cada clase de equivalencia de la salida correcta. Luego se calcula el total, a partir de cada una de las clases.

Resumiendo lo presentado en [19], para cada S_i clase de equivalencia de la salida correcta se calculan los siguientes valores:

- $c(S_i) = |S_i| - 1$ representa la cantidad mínima de vínculos requeridos para representar correctamente a S_i.
- $p(S_i)$ es un conjunto que contiene la unión de todos aquellos R_j incluidos en S_i. Además de estos R_j, eventualmente contiene menciones aisladas que están en S_i pero no se encuentran en ningún R_j.
- $m(S_i) = |p(S_i)| - 1$ representa el número de vínculos faltantes en la respuesta del sistema.
- Finalmente se calcula el recall como el cociente entre la cantidad de vínculos realizados correctamente, y la cantidad de vínculos necesarios para representar correctamente la clase S_i:

$$\frac{c(S_i) - m(S_i)}{c(S_i)}$$

  .

- Para calcular la precisión, se intercambia el conjunto salida con el conjunto respuesta y se procede de la misma manera.

Comentarios respecto a la métrica MUC-6

La principal desventaja de esta métrica es que la detección correcta de menciones individuales (sin correferencias) no es considerada, problema que originó proponer algunas nuevas, como la métrica B^3.

Otra desventaja es que todos los errores son considerados de igual peso. Por ejemplo, un vínculo incorrecto entre dos clases de equivalencia con cinco menciones cada una, es penalizado con la misma gravedad que vincular a uno de estos grupos con una mención individual.

6.3.3 Métrica B^3

Esta métrica en lugar de basarse en los vínculos entre menciones, calcula precisión y recall para cada entidad del discurso.

El algoritmo propuesto inicialmente asume que los elementos de la salida y los elementos de la solución correcta coinciden. Esta solución inicial, es la siguiente:

Si e está incluida en la clase R_i de la respuesta del programa, e incluida en la clase K_j del conjunto solución:

- $\mathrm{P(e)} = \dfrac{\text{cantidad de menciones correctas en } R_i}{|R_i|}$
- $\mathrm{R(e)} = \dfrac{\text{cantidad de menciones correctas en } R_i}{|K_j|}$

El recall y la precisión globales se calculan a través de la suma ponderada de los valores obtenidos por cada elemento. Si tenemos N menciones en total, se calculará de la siguiente forma:

$$P = \sum_{i=1}^{N} w_i * P(e_i)$$

$$R = \sum_{i=1}^{N} w_i * R(e_i)$$

En [19] se proponen dos ponderaciones distintas; una más adecuada para la resolución de correferencias entre distintos documentos, y otra para la resolución de correferencias en contexto de recuperación de información.

También se proponen distintas variantes para penalizar las menciones que están solo en la respuesta del programa, o solo en la salida correcta:

- B^3_0 : No considera las $e \notin K$, y para las $e \notin R$ les asigna el valor 0 como valor de recall.
- B^3_{all} : Penaliza ambos casos ($e \notin K$ y $e \notin R$) asignándole el valor $\frac{1}{|R|}$ o $\frac{1}{|K|}$ respectivamente.
- $B^3_{n\&g}$: Quita las menciones individuales (sin correferencias) antes de aplicar el algoritmo B^3_{all}.
- B^3_{sys} : Re-define los conjuntos R y K según se esté calculando precisión o recall, con el objetivo de mejorar la puntuación para menciones individuales.

Comentarios respecto a la métrica B^3

Una de las desventajas que se mencionan sobre esta métrica es que las entidades (en el sentido de conjunto de menciones) son utilizadas muchas veces al calcular precisión y recall de una mención particular. Para mejorar esto, surge la métrica CEAF.

6.3.4 CEAF

El objetivo del algoritmo empleado en CEAF es calcular cuánta similaridad tienen dos entidades. Se busca alinear las entidades de la respuesta del programa con las entidades de la solución, de forma de que la similaridad de dicho alineamiento sea máxima.

La principal contra que se menciona, es que el valor de precisión está sesgado al número de clases de la respuesta del programa.

También existen variaciones $CEAF_{n\&g}$ y $CEAF_{sys}$, donde se aplican las mismas ideas sugeridas en sendas variaciones de B^3.

6.3.5 BLANC

Esta métrica es bastante reciente, fue utilizada por primera vez en el Semeval 2010 y se basa en implementar el llamado "Índice de Rand", propuesto en 1971 para evaluar métodos de agrupamiento (clustering).

Tiene como principal desventaja asumir que las menciones de la respuesta y de la solución coinciden.

6.3.6 Conclusión

Las métricas de correferencias fueron estudiadas con el fin de evaluar si alguna de ellas podía ser aplicada a los resultados de este proyecto. Se determinó que, exceptuando la medida F, en la mayoría de las métricas la adaptación al problema que aborda este proyecto no es directa y se requeriría de una investigación más profunda para hacerlo.

Esto se debe a que las métricas que han surgido buscan matizar las penalizaciones de los errores de la detección de menciones. Se recuerda que en este contexto las menciones candidatas a ser correferentes están determinadas por las fuentes encontradas en el texto. No existe la "detección de menciones" como tal, etapa que sí existe en la resolución de correferencias más general.

Por ello, y también para tener una referencia respecto a [3], este proyecto será evaluado utilizando medida F como métrica para la resolución de correferencias.

6.4 Resultados de la etapa de Resolución de Correferencias

En esta sección se exponen y analizan los resultados obtenidos en la resolución de correferencias.

Resultados

A continuación se presenta la cantidad de correferencias que contenían los textos de prueba, así como las resueltas por el sistema:

Cantidad de correferencias existentes	**Cantidad de correferencias resueltas**	**Cantidad de correferencias resueltas correctamente**
101	84	79

Obteniendo los siguientes resultados para la medida F:

Precisión	**Recall**	**Medida F**
94%	78%	85%

Análisis

Si comparamos este resultado con el mejor resultado alcanzado en [3] (F=83.2%), podemos decir que el resultado es muy bueno debido a que fue logrado sin hacer ninguna corrección sobre las salidas de los textos. El número logrado es el que posiblemente se alcance al integrar el sistema al buscOpiniones, lo cual podría ampliar el número de opiniones encontradas de un modo perceptible.

La mejora en la resolución de correferencias puede deberse a:

- Utilización de WordNet únicamente para vincular entidades, nunca para descartar vínculos. La base de datos de WordNet no contiene una cantidad suficiente de entradas como para que la ausencia de relaciones influya sobre alguna decisión.
- Confirmar vínculos al compartir un nombre propio de persona u organización.
- Descartar vínculos al encontrar que dos menciones contienen nombres propios, y éstos son distintos.
- Implementación de la distancia de Levenshtein para tolerar errores de typeo entre menciones.

En términos generales, se buscó en todo momento priorizar la precisión por sobre el recall. Esto se debe a que el sistema está pensado para ser integrado a un sistema de recuperación de información, por lo que los falsos positivos (correferencias erróneas) constituyen un error de mayor gravedad que los falsos negativos (correferencias no identificadas).

Este objetivo fue conseguido, ya que se alcanzó una precisión del 94% y un recall de 78% (a diferencia del más balanceado resultado obtenido en [3], precisión 81% y recall 85.6%).

Errores encontrados, y sus posibles causas

Como se mencionó en el capítulo anterior, para que una mención ingrese a una clase de menciones correferentes se requiere únicamente correferencia con una de las menciones de dicha clase. Esto puede inducir errores como el siguiente:

[Daniel Fernández] [Fernández] [jefe de gabinete de el gobierno argentino Alberto Fernández]

El sistema descarta la correferencia entre Daniel Fernández y Alberto Fernández, ya que son nombres propios distintos. El problema surge en que la mención "Fernández"

hace que ambos queden en la misma clase, ya que la correferencia de cualquiera de los dos con "Fernández" es clara.

Otro tipo de error que quizás sea el más difícil de solucionar, es aquel donde la correferencia es establecida a partir de conocimiento del mundo. Esto se pretende resolver a través de WordNet pero salvando contadas excepciones, está lejos de brindar el conocimiento semántico requerido para interpretar algunos artículos (lo cual por otra parte, es razonable, ya que son datos muy locales empleados en la prensa uruguaya). Ejemplo:

Grupo 1: [Tabaré Vázquez] [Vázquez]
Grupo 2: [El presidente]

Claramente "El presidente" se refiere a la entidad "Tabaré Vázquez", aunque no hay forma de deducirlo con las herramientas que dispone este sistema.

Conclusión respecto a los errores encontrados

Es viable realizar algún tipo de mejora respecto a los agrupamientos erróneos a través de no agrupar a la clase de equivalencia cuando se tiene evidencia de que definitivamente no hay correferencia con una de las menciones de la clase (ej. tienen nombres propios de persona distintos).

Pero la principal causa de los errores, y además la más difícil de abordar, es la de las correferencias que requieren conocimiento del mundo para resolverse. Como se dijo, este proyecto solo utiliza WordNet para buscar relaciones semánticas, pero evidentemente todavía no tiene suficiente información para ayudar a vincular, por ejemplo, nombres propios de persona a nombres comunes.

Quizás pueda mejorarse desarrollando una aplicación específica destinada a encontrar las entidades más mencionadas por los medios de comunicación y guardar de qué formas pueden ser referidas.

Capítulo 7

Conclusiones

Ampliando las conclusiones concretas mostradas en el capítulo de evaluación, en éste se presentan conclusiones más generales sobre lo abarcado en este proyecto.

Las expectativas de este proyecto se centraron básicamente en dos puntos importantes:

- La integración de varias aplicaciones para un procesamiento automático.
- El desempeño del módulo de resolución de correferencias, tanto en los resultados alcanzados como en la robustez del sistema.

7.1 Conclusiones generales

Se desarrolló un sistema de resolución de correferencias entre fuentes de opiniones capaz de ser utilizado directamente sobre textos de prensa.

El sistema incorpora los subsistemas que utiliza y adapta sus salidas a una base de datos, hecho que permite la fácil integración de nuevos sub-sistemas al procesamiento así como la automatización de su utilización, sin requerir intervención del usuario.

7.2 Integración entre las aplicaciones

Los primeros meses de este proyecto fueron destinados a integrar todas las herramientas realizando los scripts necesarios para nutrir la base de datos del procesamiento de las siguientes salidas:

1. Artículo de prensa (texto plano en español) codificado en UTF-8.
2. Artículo procesado por FreeLing.
3. Archivo .xml con las opiniones identificadas, corregido y validado.
4. Archivo .CoNLL con los parseos de dependencia de las oraciones del artículo.
5. Módulo de resolución de correferencias entre fuentes (guarda sus resultados en la base de datos mencionada)

Se consiguió automatizar este proceso y unificar toda la información en una base de datos, lo cual consistió en un primer objetivo importante que permitió pensar con mayor libertad la construcción de un nuevo módulo de resolución de correferencias.

7.3 Módulo de correferencias

Se desarrolló un módulo de resolución de correferencias capaz de funcionar de forma autónoma recibiendo como entrada un artículo de prensa en texto plano.

7.3.1 API

Como se comentó en el capítulo de Implementación, hay una capa intermedia entre el módulo de resolución de correferencias y la base de datos, que si bien fue desarrollada de forma ad-hoc para dicho módulo, tiene el potencial para ser utilizada en otras tareas que puedan requerir de la información de FreeLing, del módulo de opiniones, del parser de dependencias (MALT) y de WordNet[1].

7.3.2 Módulo de resolución de correferencias

El módulo implementado mostró cumplir con las expectativas que se tenía, ya que si se decidía realizar un módulo que viniera a sustituir el desarrollado en [3], debía como mínimo brindar la misma tasa de resolución y mejorar la robustez para poder integrarlo a buscOpiniones.

El sistema fue probado con 45 textos de prensa elegidos al azar y, dejando de lado cuán correctos fueran los resultados, en todo momento se mantuvo estable y devolvió la salida esperada. Con esto se logró un primer objetivo importantísimo.

Respecto a los resultados, si bien cumplieron con las expectativas, la recuperación de fuentes es el área del proyecto donde se pueden obtener mejoras más fácilmente. La calibración de las ponderaciones fue realizada sobre 20 textos de prensa. Esto permitió observar ciertos patrones en la forma de referenciar las entidades de discurso y así agregar heurísticas, tanto positivas (para encontrar la fuente adecuada) como restrictivas (dónde no buscar la fuente).

[1]Las aplicaciones mencionadas son las requeridas por el módulo de resolución de correferencias. No sería difícil agregar nuevas herramientas y las respectivas funciones en la API.

En caso de seguir explorando textos y agregando nuevas heurísticas (sobretodo restrictivas), posiblemente se mejoraría el desempeño del módulo.

Capítulo 8

Mejoras a futuro

En el transcurso del proyecto se identificaron algunas ideas que resultaron interesantes pero no fue posible realizarlas o bien por razones de tiempo, o por razones de alcance. Estas ideas son listadas en este capítulo como mejoras a futuro.

8.1 Opiniones incluidas en citas

Anteriormente se mencionó que una de las restricciones que se agregaron a este módulo fue la de no buscar fuentes o predicados de opinión dentro de una cita.

"**El ministerio** será notificado oportunamente" **agregó** el diputado.

En este caso, "El ministerio" no se tiene en cuenta como candidato al predicado "agregó". Una interesante mejora a futuro es distinguir dos (o más) niveles de alcance de las opiniones, para permitir recuperar opiniones que estén enteramente dentro de una cita textual, sin mezclarlas con las opiniones del "primer nivel de alcance".

8.2 Enriquecer Wordnet para el contexto de opiniones

Una mejora que impactaría positivamente en la resolución de correferencias es agregar a Wordnet palabras, synsets y relaciones que sean utilizadas normalmente en artículos periodísticos. En estos artículos es frecuente que se utilicen sinónimos e hiperónimos para evitar la redundancia, pero estos no siempre se encuentran en Wordnet.

8.3 Utilizar técnicas de Aprendizaje Automático

Una de las virtudes principales de tener una arquitectura modularizada es que a la hora de decidir qué grupo nominal es fuente de una cierta opinión, o qué fuentes son correferentes, se tiene rápido acceso a información de distinta naturaleza (FreeLing, Wordnet, etiquetas del módulo de búsqueda de opiniones, parser de dependencias).

Dichas decisiones son tomadas en base a reglas que aplican sobre un subconjunto de esta información del texto que estamos procesando, y estas reglas fueron definidas empíricamente a partir del estudio de las estructuras del lenguaje en el que está

escrito el texto.

Un enfoque interesante consiste en, en lugar de resolver el problema buscando el conjunto de reglas más preciso y efectivo, resolver el problema obteniendo mediante la estadística cuál es el subconjunto relevante de la información disponible y de qué forma ponderar cada atributo del subconjunto.

Para ello necesitaríamos elaborar un corpus de gran tamaño, donde cada predicado de opinión tenga su fuente marcada, y donde también estén marcadas las fuentes correferentes. Teniendo un corpus de datos con esas características, podríamos utilizar un modelo de predicción y entrenarlo para que vincule la información disponible en la base de datos con el hecho de "ser correferentes" o "ser fuente de opinión".

8.3.1 Árboles de decisión

Cabe mencionar que para este tipo de tarea puede ser de interés utilizar un modelo de predicción que nos dé una pista de cómo se llega a la decisión de que, por ejemplo, dos entidades correfieren. Si logramos construir un árbol de decisión a partir del corpus mencionado, no solo podríamos resolver con mayor o menor exactitud las correferencias de un texto, sino que podríamos encontrar nuevas heurísticas para abordar las dos tareas principales que resuelve este sistema.

Bibliografía y Referencias

[1] DATA MEDIA
http://www.datamedia.com.uy

[2] Jurafsky, Daniel y James H. Martin.
Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition. 2da edición.
New Jersey: Prentice Hall. 2008.

[3] Fernando Acerenza, Macarena Rabosto, Magdalena Zubizarreta.
"Resolución de correferencias en expresiones de opinión".
Proyecto de Grado, Facultad de Ingeniería, Universidad de la República, Uruguay, 2010

[4] Jairo Bonanata, Rodrigo Stecanella.
"Extracción de opiniones de prensa".
Proyecto de Grado, Facultad de Ingeniería, Universidad de la República, Uruguay, 2013

[5] Vincent Ng.
Supervised Noun Phrase Coreference Research: The First Fifteen Years. Human Language Technology Research Institute, University of Texas at Dallas

[6] Rosá, Aiala.
"Identificación de opiniones de diferentes fuentes en textos en español.".

Tesis de Doctorado. Universidad de la República (Uruguay) / Université Paris Ouest Nanterre La Défense

[7] Aiala Rosá, Dina Wonsever, Jean-Luc Minel.
Combining Rules and CRF Learning for Opinion Source Identification in Spanish Texts.
Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
Université Paris Ouest Nanterre la Défense, Nanterre, France

[8] Hamidreza Kobdani, Hinrich Schütze.
SUCRE: A Modular System for Coreference Resolution.
Institute of Natural Language Processing, University of Stuttgart, Alemania, 2010.

[9] Emili Sapena, Lluis Padro and Jordi Turmo.
RelaxCor: A Global Relaxation Labeling Approach to Coreference Resolution.
Universitat Politecnica de Catalunya. Barcelona, Spain

[10] Giuseppe Attardi, Stefano Dei Rossi, Maria Simi.
TANL-1: Coreference Resolution by Parse Analysis and Similarity Clustering.
Università di Pisa, Largo B. Pontecorvo. Italia.

[11] Desislava Zhekova, Sandra Kubler.
UBIU: A Language-Independent System for Coreference Resolution.
University of Bremen. Germany. University of Indiana. EEUU.

[12] Lluís Padro (Universitat Politècnica de Catalunya), Evgeny Stanilovsky (FreeLing project developer).
FreeLing 3.0: Towards Wider Multilinguality.

[13] Joakim Nivre, Johan Hall, Jens Nilsson.
MaltParser: A Data-Driven Parser-Generator for Dependency Parsing.
School of Mathematics and Systems Engineering, Vâxjô University, Suecia.

[14] Montserrat Marimon (Universitat de Barcelona), Beatriz Fisas, Núria Bel, Blanca Arias, Silvia Vázquez, Jorge Vivaldi, Sergi Torner, Marta Villegas Mercè Lorente
The IULA Treebank.
Universitat Pompeu Fabra, Barcelona, España.

[15] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller.
Introduction to WordNet: An On-line Lexical Database.
Universidad de Princeton, Estados Unidos

[16] Aitor Gonzalez Agirre, Egoitz Laparra, German Rigau.
Multilingual Central Repository version 3.0: upgrading a very large lexical knowledge base.
Basque Country University, Donostia, Basque Country.

[17] Ad Neeleman, Kriszta Szendrői.
Radical Pro-Drop and the morphology of pronouns.
Department of Phonetics and Linguistics, UCL, Gower Street, London.

[18] Jie Cai and Michael Strube.
Evaluation Metrics For End-to-End Coreference Resolution Systems.

[19] Amit Bagga and Breck Baldwin.
Algorithms for scoring coreference chains.

[20] Mariona Taule, M. Antonia Martí, Marta Recasens.
AnCora: Multilevel Annotated Corpora for Catalan and Spanish Department of Linguistics, University of Barcelona

[21] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim.
A machine learning approach to coreference resolution of noun phrases. Compu-

tational Linguistics, 27(4):521–544.

[22] Shane Bergsma and Dekang Lin.
Bootstrapping path-based pronoun resolution. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pages 33–40.

[23] Niyu Ge, John Hale, and Eugene Charniak.
A statistical approach to anaphora resolution. In Proceedings of the Sixth Workshop on Very Large Corpora, pages 161–170.

[24] Simone Paolo Ponzetto and Michael Strube.
Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In Human Language Technologies 2006: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference, pages 192–199.

[25] Aria Haghighi and Dan Klein.
Unsupervised coreference resolution in a nonparametric bayesian model. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 848–855.

[26] Hoifung Poon and Pedro Domingos.
Joint unsupervised coreference resolution with Markov Logic. In Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 650–659.

LINKS

[27] Expert Advisory Group on Language Engineering Standards
http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html.

[28] IULA 04 Soaplab Web Services
http://ws04.iula.upf.edu/soaplab2-axis/

[29] Memory based learning
https://www.cs.cmu.edu/ schneide/tut5/node9.html

[30] SWI Prolog
http://www.swi-prolog.org/

Apéndices

Apéndice A

Funciones provistas por la API

Se listan a continuación algunas de las funciones más relevantes provistas por la API:

Función	**Propósito**
obtener_tag	Permite obtener todas las palabras etiquetadas como Fuente, o GN, o Predicados que estén comprendidas en el rango que se pasa como parámetro.
obtener_palabras	Permite obtener todas las palabras del texto junto con sus etiquetas según cada aplicación externa.
subtexto_contiene_nombre_propio	Devuelve 1 si el segmento de texto pasado como parámetro contiene algun nombre propio, y devuelve 2 si contiene algún nombre propio de persona o de organización (FreeLing). Devuelve 0 en otro caso.
subtexto_contiene_subj	Devuelve true si el segmento de texto pasado como parámetro contiene alguna palabra con función sujeto en su oración. La información se obtiene del MALT parser
subtexto_es_aposicion	Devuelve true si el segmento de texto tiene las características de una aposición (primera mención de un nombre propio)
verificar_concordancia	Devuelve true si existe concordancia de número entre los dos segmentos pasados como parámetro
verificar_zona_valida	Recibe como parámetros un predicado de opinion, y un segmento que es candidato a ser fuente. Devuelve false en caso de que existan razones para descartar ese candidato (ej. está entre comillas dentro de una cita, o está dentro del asunto de la opinión)