PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

Implementación de una aplicación de detección de subjetividad en textos escritos en español

Tesis para obtener el título profesional de Ingeniero Informático

AUTOR:

Rodrigo Angelo López Condori

ASESOR:

Mg. Marco Antonio Sobrevilla Cabezudo

Lima, Julio, 2025

Informe de Similitud

Yo, Marco Antonio Sobrevilla Cabezudo, docente de la Facultad de Ciencia e Ingeniería de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado Implementación de una aplicación de detección de subjetividad en textos escritos en español, del autor Rodrigo Angelo López Condori, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 36 %. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 24/09/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- El presente trabajo posee una puntuación de 36% debido que lleva como apéndice el artículo del mismo alumno que fue publicado en una revista (páginas 69-78 de la tesis). El artículo original puede ser encontrado en el siguiente link: https://www.cys.cic.ipn.mx/ojs/index.php/CyS/article/view/3279
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: 13 de Febrero de 2025

Apellidos y nombres del asesor : Sobrevilla Cabezudo Marco Antonio	
DNI: 46299018	Firma
ORCID: https://orcid.org/0000-0001-7625-9914	

Resumen

Las empresas siempre están interesadas en las opiniones que puedan brindar los clientes respecto a sus productos y servicios, las cuales son expresadas a través de distintos medios (redes sociales, foros entre otras). Sin embargo, dada la gran diversidad de comentarios de múltiples usuarios sobre, además de que no todos estos son necesariamente opiniones, se vuelve difícil para las empresas aprovechar una posible retroalimentación. Por lo tanto, se vuelve primordial filtrar la información relevante.

El área que se encarga del análisis de esta información, es el Análisis de Sentimiento, la cual se define como el área que busca identificar opiniones, pensamientos o creencias que tienen las personas respecto a una entidad específica o un atributo de está. El Análisis de Sentimiento se puede abordar de 2 formas, realizando una división entre oraciones objetivas y subjetivas para luego definir su polaridad (positivas o negativas); o considerando el problema como clasificación ternaria, con las categorías positivo, negativo y neutro. La primera forma descrita, es en la que se enfoca este trabajo

Si bien los métodos tradicionales para detección de subjetividad obtienen buenos resultados, estos aún tienen ciertas carencias incluyendo problemas con la desambiguación de palabras y las opiniones implícitas (oraciones objetivas que guardan un valor subjetivo). Una de las causas es que la mayoría de las veces se atribuye valor (subjetivo u objetivo) a una o dos palabras, dejando de lado el sentido con el que son usadas y las relaciones entre ellas.

Es debido a esto que se propone la realización de un algoritmo de aprendizaje supervisado para la detección de subjetividad, que cubra las dificultades mencionadas, para lo cual este identificara el contexto en el cual se encuentran las palabras para luego determinar si las oraciones expresan o no una opinión.

Dedicatoria

A mis hermanos Ronald y Andrea, a mi abuelita Rebeca a mi tío Luis y a mi madre Gladys, por haberme soportado todos estos años y ayudarme cada día a salir adelante y poder lograr mis metas.

Agradecimientos

Agradezco a mis hermanos Ronald y Andrea, a mi abuelita Rebeca a mi tío Luis, a mi madre Gladys y a toda la familia que siempre está brindándome su apoyo y siempre espera lo mejor de mí. Gracias por todo su apoyo y por ayudarme a perseverar en mis metas, sin ustedes no hubiese llegado tan lejos.

Agradezco a los pocos amigos que tengo por las experiencias vividas y/o cualquier forma de apoyo que me hayan brindado.

Agradezco al grupo ACM-ICPC PUCP, por haber hecho más interesante mi estancia en la universidad y brindarme quizás de las mejores experiencias que he vivido.

A mi asesor Marco Sobrevilla, por brindarme su conocimiento y su paciencia en el desarrollo de este trabajo, sin su ayuda quizás nunca hubiese realizado algo así.

Agradezco al grupo IA-PUCP por todo el apoyo brindado, sobre todo al Dr. César Beltrán y a mi buen amigo Franco Pariasca por su apoyo en la presentación de este trabajo y en general por toda su ayuda.

Agradezco también a quien se tome la molestia de revisar este trabajo, espero sea de utilidad.

Gracias a todos.

Gracias Totales

Índice

Índice de Figuras

Índice de Tablas

Generalidades

1. Problemática

Las empresas siempre están interesadas en las opiniones que puedan brindar los clientes respecto a los productos y servicios que estas ofrecen. Gracias al análisis de estas opiniones, las empresas pueden estar al tanto de los gustos e intereses que tienen los clientes y orientarse a ellos con el fin de maximizar sus ganancias y la satisfacción del cliente (He et al., 2016).

Con el avance de la tecnología se ha ido facilitando la obtención de las opiniones y comentarios de clientes, en gran medida gracias al apoyo de la Internet o de manera más específica de la Web 2.0 (Baur, 2016), siendo esto posible mediante redes sociales, *blogs*, encuestas disponibles en portales asociados a las propias empresas o en general al área de estas; refiriéndose a esto de manera más general como sitios web que faciliten compartir información.

Si bien se ha mencionado la facilidad y el beneficio de la obtención de opiniones existentes para las empresas que necesitan de información relevante, hay que examinar esto desde otra perspectiva, puesto que el análisis de este tipo de opiniones (sin ningún tipo de filtro) representa un gran problema. Usualmente las personas expresan y comparten muchas de sus emociones, sea con un grupo pequeño de conocidos o amigos, o con un grupo aún más grande de personas utilizando los diversos medios que ofrece la Web 2.0 como Twitter o Facebook (Bazarova et al., 2015). Esta información es abundante pero no es de mucha importancia para las empresas, además de esto la extracción de información resulta más complicada debido a la inclusión de diversos elementos como, errores gramaticales, lenguaje coloquial, entre otros (Petz et al., 2013).

A continuación, se presentan algunos ejemplos de posibles expresiones de cualquier persona, opinando sobre algún producto o servicio en general:

1. *Este celular es una basura, de lo PEOR, odio esta empresa y odio mi vida!!!!!* (Opinión de una persona regular)

2. *Mi dispositivo móvil me ha estado dando problemas últimamente debido a la duración de su batería, la cual es muy corta.* (Opinión mejor pero muy extensa, por lo que tiene información innecesaria)

3. *La batería de mi celular es muy mala, siempre falla* (Opinión más infrecuente, que va directo al punto)

Como se observa en los ejemplos, existe una gran variedad de posibles opiniones de los usuarios, de las cuales difícilmente se puede obtener la información que se desea. No solo esto, cuando se hace el análisis de opiniones se revisan miles de opiniones de distintos usuarios sobre distintos productos o servicios, por lo que hay una necesidad de filtrar la información relevante para la empresa (por ejemplo, las opiniones) de la que no lo es. También, de acuerdo con Liu (2012) un humano promedio tendría problemas tanto para identificar los sitios web con información relevante como para la extracción y la sumarización de esta información; es debido a esto que sistemas automatizados de análisis de sentimiento son necesarios.

Análisis de Sentimiento es definida como el área que busca identificar las opiniones, pensamientos o creencias que tienen las personas en relación a una entidad específica o a un atributo de está (Liu, 2012). Siendo este el encargado del análisis de este tipo de información, en la cual nos estamos enfocando, es primordial el entender la manera en que estudia y/o da solución al problema descrito.

La resolución del problema de Análisis de Sentimiento puede variar dependiendo del nivel de análisis que se utilice. Entre estos niveles se encuentran: (1) el análisis a nivel de documentos, el cual considera la opinión de un documento en su totalidad sea

positiva o negativa; (2) el análisis a nivel de oraciones, el cual abarca la estructura de las oraciones buscando las posibles opiniones que estas contengan; y (3) el análisis a nivel de aspectos, que busca identificar las opiniones existentes enfocándose en los aspectos y/o características de las entidades descritas en un texto. (Liu, 2012).

Específicamente, el Análisis de Sentimiento a nivel oraciones puede ser abordado de 2 formas: (1) realizando una división entre oraciones objetivas y subjetivas y luego determinando la polaridad de una opinión (positiva o negativa) y (2) considerando el problema como si fuera de clasificación ternaria, esto es, clasificando en positivo, negativo y neutro (Liu, 2012).

La primera forma mencionada en el párrafo anterior es en la que se enfoca este trabajo, específicamente, en la primera etapa, la cual es llamada detección de subjetividad. Detección de Subjetividad es definida como la clasificación de un texto en 2 categorías: objetiva o subjetiva. De acuerdo a Liu (2012), este problema es considerado un problema aparte y que no está necesariamente asociado al análisis de sentimiento, pero menciona que algunos autores lo consideran como un paso previo al asignar una polaridad (valor positivo o negativo) a alguna opinión.

Respecto a la segunda forma descrita es importante en el análisis de sentimiento ya que, clasificar los textos directamente como solamente positivo y negativo (clasificación binaria), puede desencadenar errores; ya que para esto debe asumirse que ya todos los textos guardan una opinión (Bouazizi & Ohtsuki, 2016). Por otro lado, Mihalcea et al. (2007) afirman que el problema de distinguir entre oraciones subjetivas y objetivas es más difícil que la clasificación de la polaridad, por lo tanto, mejoras en la detección de la subjetividad podrían contribuir de manera positiva en la clasificación de la polaridad.

Los métodos actuales desarrollados han mostrados ser útiles en la detección de subjetividad. Sin embargo, estos aún tienen ciertas carencias y hay diversos mensajes que no pueden manejar. Por ejemplo:

4. *Mi teléfono se apaga una y otra vez.* (Opinión Implícita)
5. *El nuevo Samsung Galaxy Note 7 es la bomba.* (Oración subjetiva)
6. *Había una bomba en la escuela.* (Oración objetiva)

Con respecto al cuarto ejemplo, un clasificador tradicional podría etiquetar la oración como objetiva al no saber cómo interpretarlo y determinar que no aporta ningún valor lo cual es falso, ya que las expresiones "apagar" y "una y otra vez" guardan un mensaje subjetivo; esta dificultad es mencionada por Liu (2012) respecto a oraciones objetivas guardando opiniones. En relación al quinto ejemplo, puede verse que la palabra "bomba" hace referencia a algo bueno y, por lo tanto, nos ayuda a identificar la oración como subjetiva. Finalmente, la sexta oración emplea la misma palabra ("bomba") pero en un sentido objetivo.

Como pudo apreciarse en los ejemplos mencionados, los métodos tradicionales atribuyen el valor (subjetivo u objetivo) a una o dos palabras, dejando de lado el sentido con el que son usadas y las relaciones existentes entre ellas; además de esto Narayanan et al. (2009) afirma que una oración puede contar con *"sentiment words"* (de ejemplo menciona grande, bonito, malo) pero eso no sería suficiente para diferenciar a una oración que presente una opinión de una que no lo haga. De esta forma, pueden incurrir en errores en la detección de subjetividad y es debido a esto que surge la importancia de mejorar los métodos existentes los cuales no pueden detectar o ignoran oraciones como las mostradas previamente.

En este contexto, surge la pregunta ¿De qué forma se puede realizar un método para la detección de subjetividad que permita cubrir todas las dificultades encontradas y descritas previamente? Es debido a esto que el presente trabajo de final de carrera propone un algoritmo de aprendizaje supervisado para la detección de subjetividad que identifique el contexto en el cual se encuentran las palabras para luego determinar si las oraciones expresan o no una opinión.

2. Marco Teórico

En la presente sección se describirán algunos de los conceptos para tener un mejor entendimiento de la problemática expuestas en el capítulo anterior.

2.1 Análisis de sentimiento

Es un área de estudio que se centra en el análisis de opiniones y sentimientos de personas hacia diversos elementos con los que entran en contacto, tales como productos, servicios, organizaciones entre otros. (Liu, 2012)

Esta área estudia las opiniones textuales expresadas por distintas personas, que pueden denotar sentimientos positivos o negativos con algún grado de intensidad, además de estar divididas bajo distintos criterios.

2.2 Tipos de opiniones

Según Liu (2012) las opiniones pueden ser agrupadas de la siguiente manera:

- Opiniones regulares y opiniones comparativas

De la opinión regular se dice que consta de 2 subtipos siendo los siguientes:

- Opinión directa: La opinión se refiere directamente a un objeto o alguna de sus cualidades. Por ejemplo: La batería de mi laptop es muy duradera
- Opinión indirecta: La opinión no está asociada directamente con el objeto en cuestión, sino con los efectos que tiene sobre su entorno. Por ejemplo: Mi nuevo sillón me causó un fuerte dolor de espalda.

Mientras que la opinión comparativa expresa diferencias o similitudes entre las entidades mencionadas por la persona que se expresó y una preferencia por alguna de estas. Por ejemplo: Mi nueva cámara digital tiene mejor resolución que la cámara de mi celular.

- **Opiniones explicitas y opiniones implícitas**

Una opinión explicita es una oración subjetiva usada para expresar una opinión. Por ejemplo: Me aburren los partidos de futbol. Por otro lado una opinión implícita es una oración objetiva empleada para manifestar una opinión. Por ejemplo: Podría ver esta película una y otra vez.

2.3 Nivel de Análisis

De acuerdo a Liu (2012) los niveles de análisis de sentimiento se pueden dividir de acuerdo a su granularidad, en tres niveles los cuales se mencionan a continuación:

- Nivel de documento

En este nivel se evalúa un documento en su totalidad a fin de clasificarlo como positivo o negativo en base a la opinión que expresa este mismo. Por ejemplo: Las opiniones que podrían darse en un *blog* sobre un determinado tema.

- Nivel de oración

Este nivel se centra en las oraciones, para determinar si es que estas expresan alguna opinión ya sea positiva, negativa o neutral (es decir que no hay opinión). Siendo este nivel el que más se relaciona con la detección de subjetividad.

- Nivel de características

Se considera este nivel como uno mucho más específico, ya que no se enfoca exactamente en estructuras de lenguaje como documentos u oraciones, sino que se enfoca en la opinión en sí misma y sobre el aspecto a la que esta se refiere. Por ejemplo, "*el juego Diablo 3 tiene muy buenas gráficas, pero posee una dificultad bastante sencilla"*. En este ejemplo se da una opinión positiva del juego evaluando una característica de este (los gráficos), sin embargo también hay una opinión negativa relacionada a otra característica (la dificultad).

2.4 Subjetividad y Detección de subjetividad

Este es un concepto clave relacionado con el análisis de las opiniones. Se entiende como subjetividad a un aspecto de lenguaje utilizado para expresar opiniones, sentimientos entre otros (Wiebe, 2004). La tarea encargada de determinar si es que una oración es objetiva o subjetiva es la clasificación, o detección, de subjetividad (Wiebe & Riloff, 2005).

Según Liu (2012) se puede considerar para la solución del problema de detección subjetividad el uso de 2 tipos de métodos:

- Basados en aprendizaje supervisado

Estos métodos implican el análisis de un conjunto de datos para crear modelos de clasificación que pueden estar basados en conceptos como el bayesiano ingenuo o el de máquinas de vectores de soporte.

- Basados en aprendizaje no supervisado

Estos métodos utilizan patrones sintácticos para la identificación de subjetividad. Para esto se emplean diccionarios de palabras o bases de datos que ya tienen valores asociados de subjetividad.

Además de dichos métodos se puede considerar lo siguiente:

- Subjetividad a nivel cross-lingüístico

Es decir, aplicar detección de subjetividad a documentos en diversos idiomas. Esto se debe principalmente al hecho de que la mayoría de estudio y desarrollo de herramientas hecho en el área del problema están disponibles en inglés. De acuerdo a Liu (2012) las empresas requieren saber las opiniones de sus clientes en cuanto a sus productos y requieren de herramientas disponibles en su idioma ya que existen pocos recursos en

idiomas que no sean inglés. La mayoría de trabajos relacionados a este tema se enfocan en análisis de sentimiento a nivel de oraciones y a nivel de documentos.

2.5 Desambiguación de sentidos de palabras (Word Sense Disambiguation o WSD)

Según Navigli (2009), la desambiguación del sentido de las palabras o WSD (por sus siglas en inglés) es la habilidad para determinar computacionalmente cual es el sentido de una palabra está siendo usado en un determinado contexto.

Este concepto es muy importante para la detección de subjetividad, ya que se precisa saber o tener entendimiento cual es el sentido que expresan las palabras y/o todo el texto analizado en sí para poder determinar si este expresa subjetividad u objetividad; esto es debido a que la misma palabra puede ser interpretada de diversas formas lo cual complica el entendimiento de la idea que trata de expresar un texto.

La desambiguación de palabras en conjunto con la detección de subjetividad suele ser estudiados juntos en lo que Liu (2012) describe como desambiguación del sentido subjetivo de la palabra o SWSD (por sus siglas en inglés *Subjectivity Word Sense Disambiguation*). Esta es la tarea que permite determinar automáticamente que palabras en un corpus están siendo usadas con un sentido subjetivo y cuales con un sentido objetivo.

3. Estado del Arte

3.1 Metodología de Búsqueda realizada

La metodología que se utilizó para el estado del arte fue la revisión sistemática (Kitchenham & Charters, 2007). Para ello se formularon algunas preguntas de investigación que dirigen este trabajo. A continuación, se describen las preguntas:

¿Qué métodos/algoritmos/técnicas existen para detectar opiniones/subjetividad?

De esta pregunta general se crearon 2 más específicas

- ¿Estos métodos como abordaron la opinión implícita?
- ¿Qué métodos incluyeron desambiguación?

Con estas preguntas construimos la siguiente cadena de búsqueda de la siguiente manera:

("method" OR "technique" OR "algorithm") AND ("subjectivity detection" OR "subjectivity classification")

El año mínimo de búsqueda fue el 2011 debido a que se quería evaluar cómo está el aprendizaje de temas de algoritmos durante los últimos 5 años. Los repositorios consultados fueron los siguientes: *Google Scholar*[1]. Al ejecutar esta cadena de búsqueda se obtuvieron 1170 resultados en *Google Scholar*.

A pesar de la gran cantidad de resultados (al menos en el primer repositorio), no todos ellos corresponden o fueron de utilidad para responder las preguntas. Por lo tanto, se realizó un filtrado de los mismos siguiendo algunos criterios de exclusión e inclusión que son detallados a seguir:

- Mínimo de 5 páginas (artículos largos) para ser aceptado
- No se aceptarán encuestas, ni tutoriales, ni informes técnicos, ni mucho menos literatura gris.
- Que la información responda a las preguntas de investigación en cuestión.

[1] Disponible en https://scholar.google.com.pe/ Accesado el 26 de Setiembre de 2016

Después de realizar el filtro de documentos, se seleccionaron 7 trabajos que serán detallados a continuación.

3.2 Resultados de la búsqueda

En esta sección se describe cómo es que los trabajos encontrados en el estado del arte, ayudaron a responder a las preguntas de investigación.

Respecto a la primera pregunta de investigación se encontraron trabajos con diversos enfoques para detección de subjetividad, como por ejemplo métodos basados en lenguaje como propone Karimi et al., 2016, también en Khanna & Shiwani, 2013 se describen 2 métodos a utilizar para detección de subjetividad, siendo uno de ellos utilizar orientación semántica y el otro enfocado solo en la subjetividad de cada palabra; cabe resaltar que se experimentó con ambos métodos por separado, pero en dicho trabajo se concluyó que usando ambos métodos juntos (uno luego de otro) se obtenía mejores resultados. Por otro lado, existen también, métodos con enfoques múltiples lenguas como se ve en Chaturvedi et al., 2015 para detección de subjetividad. Además de esto los métodos usados incluían métodos basados en reglas como en el caso de Chaturvedi et al., 2015, también métodos de aprendizaje supervisado y semisupervisado, como en los casos vistos en Switzer et al., 2011 y Karimi et al., 2016, siendo el último mencionado un trabajo que incluye ambos métodos de aprendizaje. Por último, en los trabajos se muestra que no solo importa el enfoque o el método de aprendizaje que estos empleen, además de esto se debe considerar como están estructurados los datos con los que se trabaja, que información pertinente se puede obtener de esa y que inconvenientes se puede tener por trabajar con los mismos.

Esto se puede ver en Sixto et al., 2016 que obtiene sus datos de twitter y describe como es trabajar con esta información, la cual presente limitaciones (como el número de caracteres) y también presenta información útil además del texto escrito como la

ubicación desde que se publicó o la fecha. Por otro lado en Switzer et al., 2011 la información se obtiene de reportes en el contexto de aviación, la cual es descrita como compleja. También se menciona en Khanna & Shiwani, 2013 que se usa la información de críticas de películas, en el cual resaltan el hecho de que dicha información estaba dividida en 2 partes, una enfocada en aspectos más técnicos como actores o guión, mientras que la segunda parte se centraba más en que sentía la gente respecto a la película; siendo esto algo que ayudaba a la detección de subjetividad, ya que se puede centrar el análisis en las oraciones de segundo tipo. La importancia de esto se debe a que dichos trabajos al enfocarse en datos que siguen un determinado orden, se puede afirmar que funcionan adecuadamente mientras se usen datos que sigan la estructura respectiva con la que trabajan; mientras que si se evaluasen utilizando texto de manera más general no se obtendrían los mismos resultados.

Siguiendo con las preguntas más específicas, empezando con la primera, en los trabajados revisados no se hace tanta mención a lo que es opiniones implícitas, los trabajos revisan aspectos que ya fueron descritos previamente como sus enfoques o los datos empleados, pero las opiniones implícitas son una limitación lo cual podría resultar en errores para la detección adecuada de subjetividad. Respecto a la segunda pregunta específica, se encontraron algunos trabajos en el estado de arte que sí hacían inclusión de desambiguación de palabras como por ejemplo Ortega et al., 2013 y Sobrevilla-Cabezudo et al., 2015. En ambos trabajos describen los recursos empleados (como la WordNet y la SentiWordNet) para implementar la desambiguación de sentidos de palabras y como se describe en el trabajo de Ortega et al., 2013, el incluir desambiguación mejoro significativamente el desempeño de su algoritmo para detección de subjetividad por lo que el considerar la desambiguación sería un factor importante si es que se quiere obtener mejoras en la detección de subjetividad.

3.3 Conclusiones

De toda la información brindada por los textos analizados se pudo concluir lo siguiente: Existen muchos métodos para la detección de subjetividad todos con diversas características que han sido abordados con el paso del tiempo. Sin embargo, respondiendo las otras preguntas de investigación se vio que no muchos de estos hacen mención alguna de opiniones implícitas y en su mayoría no trabajan con estas, de la misma forma no muchas incluyen desambiguación del sentido de las palabras.

De lo previamente mencionado se puede afirmar que, si bien el tema de detección de subjetividad es uno muy abordado, puesto que se le ha dado mucha relevancia en el contexto actual, aún no se ha podido explorar a fondo gran parte de los diversos aspectos que esta incluye. Esto apoya la idea que plantea el presente trabajo y refuerza la importancia de este, ya que, si bien el tema podría parecer uno ya muy recurrente, hay pocos trabajos que empleen la desambiguación del sentido de palabras y mucho menos trabajos que incluyan el análisis de opiniones implícitas y en menor medida trabajos que se orienten al idioma español, ya que la mayoría de conjuntos de datos disponibles para el procesamiento de textos se encuentra en idioma inglés. Además de esto los trabajos expuestos mostraron fallos y no ser muy exactos en la mayoría de sus casos, siendo esto relacionado con el tipo de datos con los que trabajaron, pero en una mayor medida con las características que no pueden clasificar y tienden a ignorar, como las opiniones implícitas, las que se abordaran en este trabajo con el fin de bridar una posibilidad de mejora a los métodos existentes en detección de subjetividad.

Por último se puede concluir que las herramientas que podrían ayudar con este enfoque serían las más recurrentes en los trabajos analizados entre estas el SentiWordNet , que sería muy útil mantener las frases como estructuras para analizar ya que mantienen en mejor medida el contexto y que para los datos que se usaran podría ser útil los datos

MPQA, también se ha visto que se tendrá que adaptar algún conjunto de datos del inglés al español y que además de esto el resultado de nuestro método propuesto podría depender altamente de los datos que se empleen y que tantas características se consideren.

Objetivos y alcance

1. Objetivos

1.1 Objetivo General

Implementar una aplicación de detección de subjetividad para textos escritos en español, la cual se encargará de procesar el texto para después poder determinar si estos expresan o no una opinión.

1.2 Objetivos Específicos

- Realizar la anotación de las palabras y/o expresiones de un corpus de texto a nivel subjetivo.
- Implementar un algoritmo para detectar palabras y/o expresiones que denoten subjetividad.
- Implementar un algoritmo de clasificación que permita distinguir entre oraciones subjetivas y oraciones objetivas.

1.3 Resultados Esperados

Para el objetivo específico 1:

- Esquema de anotación de las palabras en el corpus.
- Corpus anotado con subjetividad a nivel de palabras y/o expresiones.

Para el objetivo específico 2:

- Algoritmo de detección de subjetividad a nivel de palabras y/o expresiones.
- Diseño del algoritmo de detección de subjetividad a nivel de palabras y/o expresiones.
- Reporte y análisis de resultados del algoritmo de detección de subjetividad a nivel de palabras.

Para el objetivo específico 3:

- Modelo de clasificación de oraciones subjetividad y objetivas.
- Características a ser consideradas en el modelo de clasificación de oraciones subjetivas y objetivas.
- Reporte y análisis de pruebas de validación del método de detección de subjetividad.

2. Herramientas, Métodos, Procedimientos y Metodologías

En esta sección se describirán las herramientas, métodos y metodologías más importantes que se utilizarán para el desarrollo de este proyecto de fin de carrera. Se presenta como se usará cada herramienta y/o método de acuerdo al resultado esperado en la Tabla 1:

Resultado Esperado	Herramienta, Método, Procedimiento o Metodología
Esquema de anotación de las palabras en el corpus	Elaboración de esquema de anotación de corpus KDD
Corpus anotado con subjetividad a nivel de palabras y/o expresiones.	Corpus *FilmAffinity* *KDD* Anotación de corpus con subjetividad a nivel de palabras
Algoritmo de detección de subjetividad a nivel de palabras y/o expresiones.	Python POS-Tagger WordNet-Pr Lematizador SentiWordNe Parser Sintáctico MCR 3.0
Diseño del algoritmo de detección de subjetividad a nivel de palabras y/o expresiones	Python Algoritmo basado en grafos
Reporte de resultados del algoritmo de detección de subjetividad a nivel de palabras	Extracción de medidas de *precision* y *recall.*
Modelo de clasificación de oraciones subjetividad y objetivas.	Python KDD Scikitlearn Algoritmo de clasificación
Características a ser consideradas en el modelo de clasificación de oraciones subjetivas y objetivas.	Python Scikitlearn KDD
Reporte de pruebas de validación del método de detección de subjetividad.	Extracción de medidas de *precision* y *recall.* *KDD*

Tabla 1: Relación de las herramientas con los resultados esperados

2.1 Herramientas

- Corpus

Se disponen de un corpus de datos que se empleará junto con el método propuesto, el cual es descrito a continuación:

- **Corpus *FilmAffinity*:** Este corpus está constituido por oraciones y fragmentos de texto obtenidos de *FilmAffinity*[2], el cual es un sitio web con información sobre películas, en su versión en español (Sobrevilla-Cabezudo et al., 2015). El corpus se encuentra compuesto por 2500 oraciones subjetivas y 2500 oraciones objetivas.

- Python

Este es uno de los lenguajes de programación más populares debido a la gran utilidad que posee respecto a diversas áreas de ciencias de la computación. Esto debido que esta incluye una gran variedad de librerías asociadas a dichas áreas, como por ejemplo librerías con algoritmos de aprendizaje de máquina (Pedregosa et al., 2011). De estas librerías se usará específicamente Scikit-learn para el algoritmo de aprendizaje supervisado que se desarrollará en este proyecto.

- WordNet-Pr

Según Miller (1995) Es una base de datos lexical que contiene sustantivos, verbos, adjetivos y adverbios organizados en un conjunto de sinónimos que representan el sentido de una palabra. Se usará para encontrar el sentido adecuado de las palabras.

- SentiWordNet

[2] Disponible en https://www.filmaffinity.com/es/main.html Accesado el 26 de Octubre de 2016

La SentiWordNet es un recurso léxico que aplica un criterio de clasificación sobre los sentidos de las palabras o frases existentes en la WordNet, asignándoles un valor positivo, negativo o neutro, de acuerdo a lo que corresponda (Baccianella et al., 2010). Una vez determinados los valores, se determinará si dicha palabra contiene subjetividad o no, esto servirá para poder determinar el valor de la oración en conjunto sea como objetiva o subjetiva.

-*Parser* Sintáctico

Un *parser* o analizador sintáctico es una herramienta que sigue reglas de gramática de un lenguaje para asignarle una estructura a las oraciones.

El análisis sintáctico se puede dar de dos formas distintas; siendo la primera el análisis de constituyentes, el cual se basa en descomponer la oración en componentes pertenecientes a categorías nominales, verbales, entre otros. Mientras que la segunda, y la que se usara en este trabajo, es el análisis de dependencias o relaciones gramaticales, el cual se enfoca más en las relaciones gramaticales existentes entre los componentes como el sustantivo con el determinante o el verbo con el objeto directo.

Existen diversas herramientas que incorporan un *parser* sintáctico de entre las cuales se decidió usar *FreeLing*[3] ya que se realizó comparaciones con otras herramientas existentes, como por ejemplo *UDPipe*[4] que no cubre lo requerido para el presente trabajo; o también *HISPA* [5] el cual es bastante completo, pero cobra la licencia, a diferencia de *FreeLing* que se encuentra disponible de manera gratuita.

Esta herramienta se utilizará para obtener las relaciones existentes entre cada componente de una oración que se esté analizando, además de que Freeling también incorpora otras

[3] Disponible en http://nlp.lsi.upc.edu/freeling/node/1 Accesado el 11 de Noviembre de 2016
[4] Disponible en https://ufal.mff.cuni.cz/udpipe#introduction Accesado el 11 de Noviembre de 2016
[5] Disponible en http://visl.sdu.dk/visl/es/parsing/automatic/ Accesado el 11 de Noviembre de 2016

funcionalidades como el POS-Tagger y un lematizador las cuales también se usaran en conjunto con el parser sintáctico.

- POS Tagger

El *part of speech tagger* es una herramienta para la clasificación de las palabras en un texto de acuerdo a su categoría gramatical o a su categoría sintáctica, por ejemplo, sustantivo, verbos, adverbios entre otros. Existen diversas herramientas que cumplen la esta funcionalidad, como el POS tagger de la universidad de Standford propuesto en Toutanova et al. (2003). Para este trabajo se utilizará el POS tagger de *FreeLing* para la clasificación de las palabras en el corpus.

- Lematizador

Según Díaz (2005) Un lematizador es un programa que trabaja sobre un corpus de textos y realiza una extracción de los términos simplificados en sus respectivos lemas, las cuales son la unidad mínima de significado de las palabras o también descrito como la parte esencial. En el trabajo se usará el lematizador que incorpora *FreeLing* para poder obtener los lemas del corpus respectivo.

-Multilingual Central Repository (MCR)

El *MCR*[6]es un repositorio que contiene *WordNets* de lenguas europeas, entre las cuales se encuentran disponibles el inglés y el español, además de ser de uso completamente libre. Éste se usará para obtener los sentidos de cada palabra que se encuentren en la oración y obtener los vértices del grafo sobre el que se ejecutará el algoritmo.

2.2 Métodos, Procedimientos y Metodologías

- Revisión Sistemática

[6] Disponible en http://adimen.si.ehu.es/web/MCR Accesado el 11 de Noviembre de 2016

Esta metodología se utilizó para la búsqueda de trabajos relacionados y el desarrollo del estado del arte (Kitchenham & Charters, 2007). Pero no se utilizó la metodología en su totalidad sino solo algunas partes como las preguntas de investigación.

- Esquema de anotación de palabras del corpus

Se requiere aportar más información al corpus de datos empleados, ya que, para verificar la adecuada detección de subjetividad en el corpus, se debe tener en cuenta que sentido está usando la palabra o expresión en el contexto en que está presente y su valor de subjetividad. Debido a esto es que se precisa anotar cada palabra (con su respectivo lema y categoría gramatical), su sentido correspondiente, su valor de subjetividad e indicar a que oración pertenece dicha palabra.

- Anotación de corpus con subjetividad a nivel de palabras

Se requiere un análisis de textos para detectar subjetividad en los corpus, examinando a detalle que componentes, sean estos frases o palabras, aquellos que determinen el valor objetivo o subjetivo del texto. Para esto se definirá un esquema para la identificación de subjetividad en palabras y/o expresiones, clasificándolas como objetivas o subjetivas.

- Extracción de medidas de *precision y recall*.

Para evaluar los algoritmos de clasificación y detección de subjetividad se usarán las medidas de *precision* y *recall*. La medida *precision* calcula la relación de las instancias de una clase correctamente etiquetadas respecto a las instancias que la aplicación etiquete con esa clase. Por otro lado, el *recall* calcula relación entre las instancias de una clase etiquetada respecto a las instancias que realmente pertenecen a la clase etiquetada.

- *Knowledge Discovery in Databases (*KDD)

Es un proceso iterativo para la extracción de conocimiento útil proveniente de algún repositorio de información (Brachman & Anand, 1996). Este modelo se empleara en el desarrollo del proyecto a fin de llevar una planeación y mejor manejo del mismo.

De este proceso se considerarán algunas fases, como las siguientes: Selección de datos, Preprocesamiento, Transformación, *Data Mining*, Interpretación y Evaluación. Para la fase de selección de datos, se revisó algunos corpus de datos y se decidió utilizar el corpus descrito en esta sección (*FilmAffinity*), para realizar el análisis.

Respecto al preprocesamiento, se estructurarán los sentidos de las palabras de cada oración en un grafo respectivo para poder trabajar con el algoritmo a implementar para la detección de la subjetividad de esta. Luego de esto se procederá con la fase de transformación mediante la cual se analizarán las relaciones existentes entre cada palabra de la oración, obtenidas a partir de las aristas del grafo previamente descrito. Estas relaciones serán usadas para determinar la categoría de la oración como subjetiva u objetiva.

En la siguiente etapa, se experimentarán con algoritmos de clasificación provistos por la herramienta scikit-learn de Python para poder seleccionar el clasificador que genere el mejor modelo. Posteriormente, se emplearán algunos algoritmos de selección de *features* para poder descartar las *features* que no contribuyan al aprendizaje.

Por el último, en la etapa de interpretación y evaluación se utilizarán medidas de precision y *recall* para determinar la correctitud de la información final resultante.

- Algoritmo basado en grafos

Para la implementación de este algoritmo se realizará una adaptación del algoritmo de *PageRank* (Page et al., 1999). Este algoritmo comúnmente es usado para clasificar páginas web de acuerdo al posible interés de las personas, como ejemplo se podría decir que es usado por el buscador Google. El algoritmo trabaja contando la cantidad de aristas para llegar a cada vértice del grafo, bajo la suposición de que los vértices más importantes estarán conectados con una mayor cantidad de aristas. Para este caso, se van a detallar los vértices y aristas del grafo sobre el que se ejecutará el algoritmo.

Para construir el grafo se utilizará el *parser* sintáctico para estructurar adecuadamente las oraciones o frases que se usen en el corpus. Un ejemplo de esta estructuración puede verse en la Figura 1.

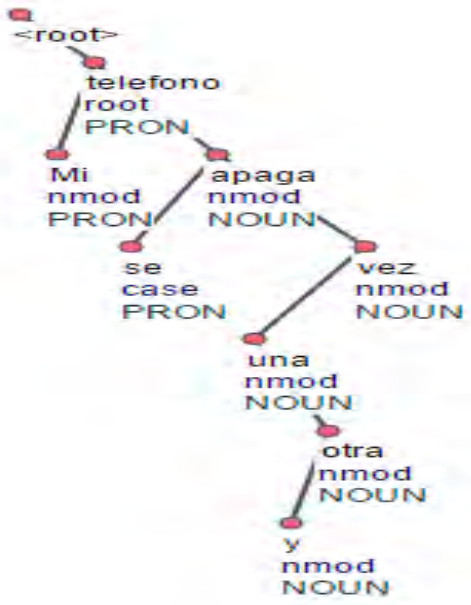

Figura 1: Representación de la oración "Mi teléfono se apaga una y otra vez" en un árbol.

A partir de esta estructura se obtendrán los vértices del grafo, que serán proporcionados a partir de las palabras y/o frases, como por ejemplo, "una y otra vez", la cual se encuentra separada, pero se unirá en un solo elemento luego de obtener la estructura.

Una vez obtenidas las palabras o frases de las oraciones con sus respectivas categorías gramaticales, se usará el MCR para obtener todos los posibles sentidos de cada palabra o frase, siendo estos los vértices del grafo que se empleará. Las aristas se obtendrán a partir de la Figura 1 a partir de las relaciones de cada palabra entre sí, enlazando cada sentido de cada palabra con los de aquellas que tengan una relación de acuerdo a la estructura mostrada. Es con este grafo que se implementará el algoritmo de PageRank, el cual se aplicará sobre el grafo y determinará el sentido apropiado de cada componente de la oración. Después, se asignará el valor de subjetividad u objetividad de acuerdo al sentido determinado; esto se logra haciendo uso del SentiWordNet que se encuentra alineado con la WordNet, es decir el MCR, dando así lo necesario para poder detectar la posible subjetividad existente en la oración con el algoritmo posterior.

- Algoritmo de clasificación

El algoritmo de clasificación será implementado usando la librería de Python Scikit-learn luego de haber obtenido los resultados esperados del algoritmo basado en grafos. Para esta clasificación, se considerarán como características o *features* no las palabras (como se suele hacer en algunos trabajos) ya que podría consumir mucho tiempo si es que se llegase a incluir una cantidad excesiva de las mismas. En su lugar, se usarán las relaciones existentes entre las categorías gramaticales de cada componente del grafo (los componentes de la oración), además de su sentido detectado, sea este objetivo o subjetivo.

Algunos ejemplos de características son mostrados a continuación:

1. *Verbo Objetivo – adverbio subjetivo*
2. *Verbo Subjetivo- sustantivo objetivo*

En el ejemplo (1) puede verse que la característica está definida como la existencia (o no) de un verbo objetivo relacionado con un adverbio que haya sido clasificado como subjetivo. Un ejemplo de esta relación puede ser la oración *"Vería la película una y otra vez"*, donde el verbo "ver" y el adverbio "una y otra vez" se relacionan y, además, la palabra "ver" es clasificada como objetiva y "una y otra vez" es clasificada como subjetiva.

A partir de estas relaciones es que se entrenará y se realizaran pruebas sobre diversos algoritmos de clasificación para poder detectar la subjetividad tanto en las opiniones tradicionales, como con las opiniones implícitas. Una vez realizadas las pruebas se optará por utilizar para la aplicación aquel que obtuvo un mejor desempeño en base a las métricas a evaluar.

3. Alcance

El presente proyecto tiene como propósito el desarrollar una aplicación de detección de subjetividad para textos en español. Para esto se requiere el desarrollo de algoritmos basados en conocimiento, ya que para este tipo de algoritmos solo se requiere contar con

algunos recursos, en este caso diccionarios y/o recursos literarios para la desambiguación del sentido de las palabras.

Otro detalle importante es que este proyecto también podrá integrarse con otros componentes o sistemas asociados a minería de opiniones y procesamiento de texto.

Los objetivos de dichos algoritmos serán, en primer lugar, detectar la subjetividad en palabras y/o expresiones. A partir de esto se espera poder identificar y etiquetar aquellas palabras o expresiones que aporten valor de subjetividad de acuerdo al contexto en que se encuentran y al sentido que posee. En segundo lugar, una vez realizado esto se podrá clasificar las oraciones como objetivas o subjetivas, esto implica un análisis no solo de las palabras sino de todo el contexto general y significado de las oraciones o frases en sí, además que se plantea el poder detectar las opiniones implícitas existentes. Además de esto la información utilizada para la aplicación se limitará a texto escrito adecuadamente, ignorando problemas que se puedan encontrar en twitter o en general en texto informal.

3.1 Limitaciones

- La mayoría de trabajos están orientados o trabajan sobre textos en inglés y no hay tantos para textos en español los cuales tienen diferentes reglas.
- No existe mucha información para la detección de opiniones implícitas para textos en español, la cual se plantea como parte del alcance.
- No se cubrirán algunos aspectos o tipos de opiniones como opiniones comparativas u opiniones con lenguaje sarcástico.

3.2 Riesgos

En la Tabla 2 son presentados los riesgos que podrían afectar el desarrollo del presente trabajo de final de carrera.

Riesgo Identificado	Impacto	Medidas
Perdida de datos	Impacto Alto. Detener el proyecto y/o tener que reiniciar el desarrollo	Hacer uso de un repositorio y tener documentación adecuada
Cambio en la disponibilidad de las herramientas	Impacto Medio. Las herramientas que se usaran ya no se encuentran disponibles o se vuelven obsoletas	Tener herramientas de respaldo o buscar sustitutos de las utilizadas
Conocimiento insuficiente sobre lingüística	Impacto Medio. Para realizar un trabajo de este tipo se debe contar con el conocimiento suficiente de lenguaje	Consultar material bibliográfico de lengua española. Consultar con expertos

Tabla 2: Tabla de Riesgos

4. Justificación

Existe una gran necesidad por filtrar y seleccionar mejor la información brindada por distintas personas, por parte de diversas empresas y compañías que ofrecen variedad de productos y/o servicios; ya que para lograr la satisfacción del cliente se debe lograr entender de mejor manera que es lo que este desea. Al tener una mejor información es que se incurre en reducir costos, pudiendo maximizar las ganancias ya que se sabe en un tiempo más corto o de una manera más directa que es lo que demanda el mercado y el público consumidor. Gracias a esto se reduce el riesgo de fomentar nuevos proyectos que podrían ser rechazados abruptamente, además de que se pueden obtener nuevas propuestas basándose en cómo se sienten los usuarios de los servicios brindados.

Según Liu (2012), la detección de subjetividad es un problema más complicado que el de detección de polaridad, debido a la existencia de oraciones objetivas, que pueden expresar alguna opinión o sentimiento (es decir opiniones implícitas). Además de esto se considera a la detección de subjetividad como un paso previo para poder asignarle algún valor a las oraciones (positivo o negativo). Debido a estos motivos es que el resultado final de este

proyecto de tesis podrá brindar un soporte adecuado a otras herramientas relacionadas con el análisis de sentimiento y la minería de opinión pudiendo mejorar el desempeño de las mismas, puesto que una gran parte de errores existentes en ese tipo de herramientas es debido a una omisión o un mal desarrollo de detección de subjetividad.

Si bien existen algunas herramientas existentes que ofrecen una solución al problema descrito en este trabajo, estos tienen algunas carencias y limitaciones, además de tener una precisión no muy elevada. La solución propuesta ofrece un enfoque distinto al de los trabajos mostrados en el estado del arte y propone superar una limitación de muchos de estos, la cual es la detección de opiniones implícitas. Las soluciones existentes presentan un enfoque que emplea diccionarios de palabras solo para saber significados dejando de lado el sentido o contexto de las mismas, es decir hay pocos trabajos que realmente busquen detectar el sentido adecuado de la palabra, siendo esto algo desarrollado en la solución presentada. Además de esto, la mayoría de soluciones usa como características las palabras que forman parte de las oraciones, mientras que en el presente trabajo se propone un enfoque completamente distinto, usando en lugar de palabras las relaciones existentes entre las mismas, siendo este tipo de propuesta novedosa, ya que no se ha encontrado en la revisión de la literatura realizada.

Corpus

1. Anotación

En esta sección se describirá como se realizó el proceso de anotación de palabras o expresiones de acuerdo al corpus y los respectivos sentidos de estas, para determinar la eficiencia del algoritmo de detección de subjetividad en palabras y expresiones, a partir de los resultados de la anotación. Cabe resaltar que las únicas categorías gramaticales (y las únicas en las que se enfocaran los demás capítulos) que se anotaran son sustantivo, verbo, adjetivo y adverbio, ya que se consideran las categorías más importantes de una oración (le dan el sentido) y porque son las únicas categorías que se encuentran en los recursos utilizados como la WordNet o la SentiWordNet.

Para la anotación se tiene un documento siguiendo la siguiente estructura; primero el tipo de oración del corpus, luego la palabra o expresión, el lema, la etiqueta gramatical o *tag*, y por último el sentido de la palabra (este sentido es anotado manualmente); lo cual se puede observar en la Figura 2:

subj	este	este	D	-
subj	inspirador	inspirador	A	spa-30-01323096-a
subj	drama	drama	N	spa-30-06376154-n
subj	,	,	Fc	-
subj	mientras_que	mientras_que	CS	-
subj	trafica	traficar	V	spa-30-02244956-v
subj	con	con	S	-
subj	clichés	cliché	N	spa-30-07154046-n
subj	,	,	Fc	-

Figura 2: Anotación del corpus

Para realizar esto se emplea un corpus de partes de textos obtenidas de la página FilmAffinity[7], el cual es un sitio web con información sobre películas, en su versión en español (Sobrevilla-Cabezudo et al., 2015). El corpus se encuentra dividido en 2 partes,

[7] Disponible en https://www.filmaffinity.com/es/main.html Accesado el 26 de Octubre de 2016

la primera está compuesta por 2500 oraciones subjetivas y la segunda por 2500 oraciones objetivas. Entonces, al anotar en conjunto las oraciones de todo el corpus es necesario saber a qué oración pertenece a cada palabra y si dicha oración es subjetiva o si es objetiva. Luego se tiene la palabra el lema y *tag*, los cuales se obtuvieron durante el pre-procesamiento, de la forma que fue descrita en la sección de implementación del algoritmo para detección de subjetividad. Por último, se tiene el sentido de dicha palabra, para lo cual se usa la WordNet para el español junto con la WordNet en inglés (para casos donde se requiera, que se mencionaran más adelante), para obtener los posibles sentidos de la palabra y se anota manualmente, de acuerdo al contexto de cada palabra que sentido corresponde efectivamente.

2. Resultados

Al ser el corpus es muy extenso no se pudo realizar su anotación total, debido a esto se trabajó con la anotación del 8% de este, es decir 200 oraciones objetivas y 200 subjetivas. Los resultados de este proceso de anotación, son presentados en la Tabla 3 y en la Tabla 4.

Categoría Gramatical	Palabras o Expresiones Totales (tokens)	Palabras o Expresiones únicas (types)	Types anotados con 1 solo sentido	Types anotados con múltiples sentidos	Types anotados con sentidos de 1 sola categoría de subjetividad	Types anotados con sentidos de múltiples categorías de subjetividad
Sustantivo	2198	1346	1195	151	1296	50
Verbo	1162	702	606	96	658	44
Adjetivo	897	700	633	67	675	25
Adverbio	363	136	114	22	129	7

Tabla 3: Tablero de Anotación del corpus

Categoría Gramatical	Sentido no subjetivo (NS)	Sentido bajamente subjetivo (LS)	Sentido medianamente subjetivo (MS)	Sentido altamente subjetivo (HS)
Sustantivo	1671	194	89	244
Verbo	676	144	244	98
Adjetivo	187	49	52	609
Adverbio	94	15	17	237

Tabla 4: Tabla de Valores de Subjetividad de sentidos anotados

Respecto a estos resultados, durante el proceso de anotación se encontraron algunas dificultades para la detección adecuada del sentido de las palabras, las cuales son descritas a continuación. Para la anotación del sentido, se encontraron algunos casos en los que había sentidos muy parecidos usando solo la WordNet en español, por lo que se utilizó la WordNet en inglés para obtener más detalle de los sentidos. Sin embargo, esto no resolvió todas las ambigüedades por lo que se optó por anotar el sentido más frecuente de todos los sentidos similares encontrados; además de esto también se encontraron palabras sin un sentido en la WordNet (como nombres propios) estas fueron dejadas de lado y no se

anotó sentido alguno, ya que no lo tienen. Por otro lado, también se encontraron casos en los cuales no se encontró un sentido asociado correcto sea en la WordNet en español o inglés. Esto sucedió debido a que el lemma no estaba asociado a algún sentido apropiado para la palabra en el contexto en que estaba, en estos casos se anotó el sentido encontrado a partir de la búsqueda usando otros lemmas, que se aproximasen a la idea que la palabra representaba en su contexto. Por último algunos sentidos fueron clasificados incorrectamente, por el POS-tagger utilizado, por lo que en dichos casos, se buscó primero la categoría gramatical adecuada de la palabra y luego se le asignó el sentido correcto de acuerdo a la categoría que se consideró correcta.

Subjetividad en palabras y/o expresiones

1. Implementación del algoritmo

En esta sección será descrito el proceso de desarrollo del algoritmo para la detección de subjetividad en palabras y/o expresiones. Para esto, se estructuraron las oraciones (datos de entrada) en un grafo luego de realizar un pre-procesamiento adecuado. Luego, al tener la estructura del grafo se utilizó el algoritmo de PageRank. A continuación, se describirá el proceso utilizado para lograr la detección de subjetividad en palabras y/o expresiones:

1.1. Pre-procesamiento

Para el preprocesamiento, se removieron espacios en blanco o caracteres encontrados que no se pudieron reconocer, para luego hacer uso de la WordNet. Con la WordNet se obtuvieron los sentidos de las palabras, además de unir palabras en expresiones que individualmente no guardaban ningún valor, pero en conjunto poseen relevancia. Por ejemplo, la expresión "una y otra vez" no se encontró en la WordNet ya que se buscó palabra por palabra, mientras que al unir cada palabra de la manera "una_y_otra_vez" esta se encontró en la WordNet y se guardó su sentido. Además, se extrajo de la WordNet para cada palabra o expresión sus sinónimos, glosas y atributos de la ontología SUMO[8] (que la WordNet asocia a sus sentidos), cuyo uso será explicado en una sección posterior.

1.2. Uso de herramientas de análisis lingüístico

Para esto se utilizó la herramienta *Freeling*[9], la cual tiene diversas funcionalidades como por ejemplo un lematizador, el cual se usó en conjunto con la WordNet, para el preprocesamiento, ya que permite obtener los lemas; además de un POS-Tagger, para identificar las categorías gramaticales de las palabras y/o expresiones presentes en el texto

[8] Disponible en http://www.adampease.org/OP/ Accesado el 11 de Noviembre de 2016
[9] Disponible en http://nlp.lsi.upc.edu/freeling/node/1 Accesado el 11 de Noviembre de 2016

y por último un *parser* sintáctico. El uso de dichas funcionalidades fue necesario ya que la WordNet no guarda directamente las palabras sino que trabaja directamente con los lemas y para identificar los sentidos se debe saber la categoría gramatical a la que pertenece. El parser que se utilizó fue un parser de dependencias para poder obtener las relaciones existentes entre los componentes de cada oración procesada, es decir las relaciones como verbo-objeto directo o sustantivo-modificador directo, entre otras. Es con estas relaciones que se pudo realizar la construcción del grafo.

1.3. Construcción del grafo

Una vez obtenidos los sentidos y las relaciones de los componentes de la oración (los cuales vendrían a ser vértices y aristas, respectivamente) es que se dio inicio a la construcción del grafo. Para esto se consideraron 2 enfoques: (1) Relacionar cada sentido de las palabras o expresiones individualmente con su respectivo valor de subjetividad y (2) agrupar los sentidos de cada palabra y/o expresión en 4 categorías de acuerdo a su subjetividad (no subjetivo, baja, mediana y altamente subjetivo). Estas categorías se obtuvieron a partir de la SentiWordNet la cual asigna valores positivos o negativos a las expresiones de la WordNet, los cuales fueron considerados como subjetividad, mientras que la objetividad fue obtenida como complemento a 1 (es decir 1 – valor positivo – valor negativo). Las categorías son asignadas de la siguiente forma si la expresión tiene 0 de subjetividad es no subjetiva, si la subjetividad es <=0.25 es bajamente objetiva, si es <=0.5 es medianamente subjetiva, y si es mayor que 0.5 es altamente subjetiva. Además de usar la subjetividad de la SentiWordNet se usó el atributo *SubjectiveAssessmentAttribute* [10] de la ontología SUMO, el cual indica posible subjetividad en dicha expresión lo cual se debe considerar, es debido a esto que cualquier

[10] Disponible en http://adimen.si.ehu.eus/cgi-bin/wei/public/hierarchy.php?name=sumo&category=SubjectiveAssessmentAttribute#SubjectiveAssessmentAttribute Accesado el 11 de Noviembre de 2016

expresión que posea dicho atributo es considerada altamente subjetiva. Se probaron las 2 estructuras descritas para el grafo (agrupando los sentidos y sin agruparlos) para probar cual es la más adecuada. Una vez completo el grafo se procedió a realizar la implementación del algoritmo de PageRank para analizar sus resultados.

En la Figura 3 se puede observar un ejemplo del corpus con el cual se trabajará durante todo este capítulo:

> este inspirador drama, mientras que trafica con clichés, logra no entregar su mensaje de una manera demasiado pesada.

Figura 3: Texto subjetivo (opinión) tomado del corpus de FilmAffinity

El texto escogido primero pasa por el preprocesamiento, de la manera descrita, es decir haciendo uso del lematizador y el POS-Tagger de *Freeling* en conjunto con la WordNet lo que dio el siguiente resultado:

Figura 4: *Preprocesamiento* del texto seleccionado

Como se observa en la Figura 4, el texto seleccionado fue descompuesto en oraciones, los cuales a su vez también fueron descompuestos en palabras o expresiones (por ejemplo "una_y_otra_vez") utilizando Freeling en conjunto a la WordNet. En la Figura 4 se muestra la palabra "drama" relacionada con la información obtenida para cada sentido encontrado como resultado del preprocesamiento. Cabe resaltar que todas las palabras poseen la misma estructura. La información asociada a cualquier palabra (en este caso "drama") está compuesto de 2 partes, siendo la primera la palabra, su lema y categoría gramatical (están en inglés porque así las etiqueta el lematizador de Freeling, 'N' es *noun,* es decir, sustantivo), y la segunda contiene los sentidos asociados en la WordNet, los sinónimos de la palabra de acuerdo a cada sentido, la glosa o el significado de dicho sentido y por último el atributo asignado de la ontología SUMO a cada sentido. Cabe resaltar que algunos componentes tienen valores de la siguiente forma '-' , los cuales indican que no se posee información. Por ejemplo para la palabra drama, se puede ver su lema el cual es drama y su categoría gramatical N, que indica que es un sustantivo (*noun* en inglés) en su primera parte; mientras que en su segunda parte se ven algunos de sus sentidos, por ejemplo el sentido **'spa-30-06376154-n'** el cual está junto con los sinónimos de ese sentido como la palabra **dramaturgia**, el guión (**'-'**) en la sección donde debería ir la glosa del sentido, pero como se indicó para este caso la WordNet no tiene esa información y por último el atributo asignado al sentido por parte de la ontología SUMO, el cual es ***text*** .

Luego de haber realizado el preprocesamiento, se hizo uso del parser de dependencias para saber qué relaciones guardan las palabras del texto seleccionado:

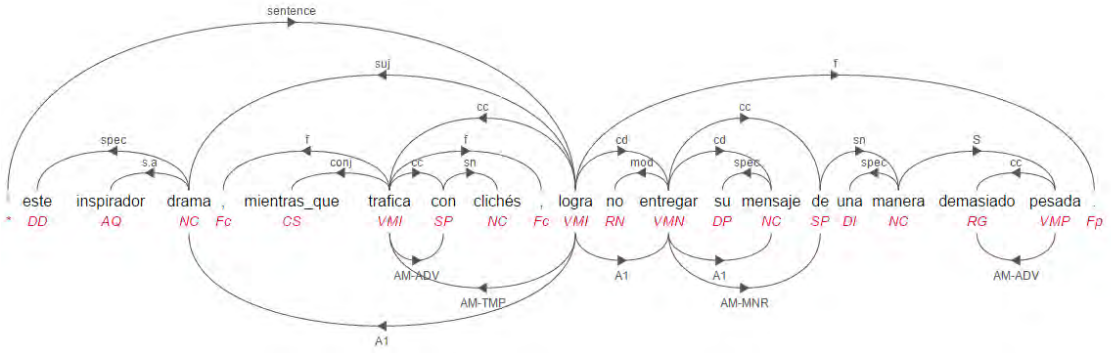

Figura 5: Relaciones obtenidas a partir del parser de dependencias

En la Figura 5 se muestran las relaciones obtenidas de cada oración obtenida a partir del parser de dependencias. Cabe resaltar que las oraciones son estructuradas como un árbol al hacer uso de dicho parser. Para poder unir todas las oraciones del texto seleccionado se optó por generar un vértice vacío que este unido a todas las raíces de cada oración, en este caso '-'.

Una vez obtenidos los sentidos y la: relaciones del parser se construyó el grafo con esa información.

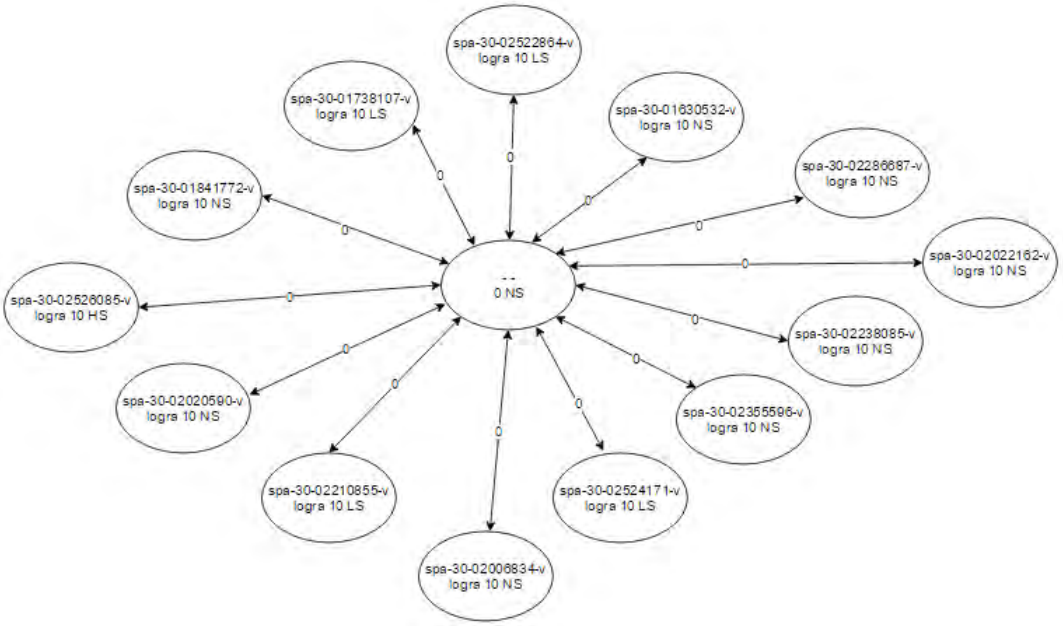

Figura 6: Grafo de sentidos (Relación entre sentidos de "logra" con el sentido vacío)

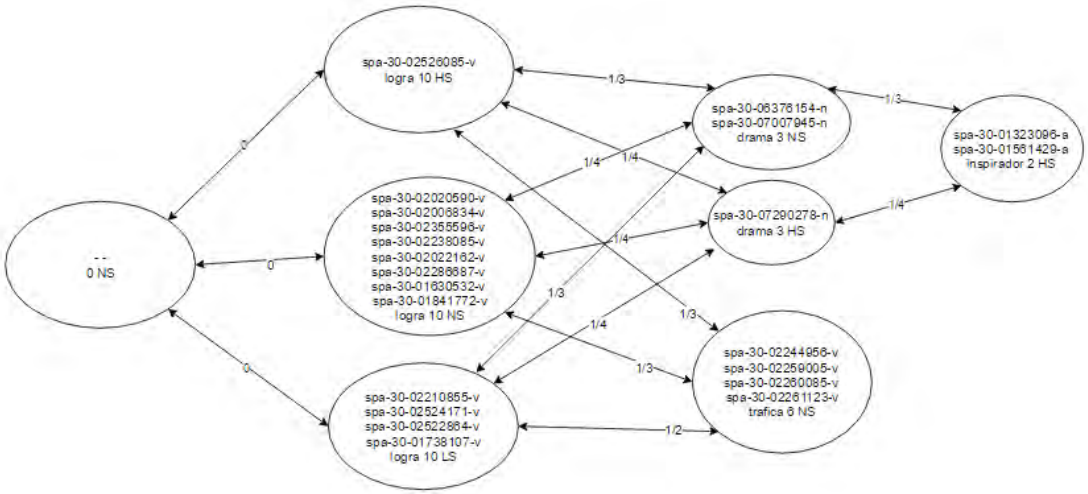

Figura 7: Grafo con sentidos unidos por subjetividad (algunas palabras)

Se muestran partes de los grafos obtenidos a partir del texto escogido (de las 2 maneras descritas previamente con sentidos separados y unidos, como se puede ver en las Figuras 6 y 7 respectivamente) debido a que no se puede mostrar ningún grafo completo al ser extenso, pero la estructura descrita en las Figuras 6 y 7 se mantiene en todo el grafo. Cada arista es descrita por el primer vértice (sentido o sentidos unidos, palabra asociada al sentido, su posición en la oración y su valor de subjetividad) el vértice con el que está conectado por la arista (descrito de la misma forma) y la distancia entre dichos vértices (el peso de la arista) cuyo valor será descrito más adelante. Como se indicó antes, se tuvo un vértice vacío (-) para unir las oraciones existentes en el texto seleccionado, aunque no fue el único posible vértice vacío ya que puede darse el caso que la raíz de la oración (representada como un árbol por el parser) no tenga un sentido encontrado en la WordNet, por lo que se optó por representarla en el grafo con el carácter '*' siendo otro vértice vacío para mantener el árbol consistente; por último si es que no se encontró algún sentido para la palabra en la WordNet y esta no era la raíz de la oración esta no fue incluida como vértice del grafo.

Para obtener las distancias entre los sentidos, durante la etapa de preprocesamiento se ejecutó el algoritmo de Dijkstra a partir de cada sentido, usando las relaciones entre sentidos que existen en la WordNet. Los resultados obtenidos del algoritmo de Dijkstra se incorporaron a las aristas del grafo, pero se usó la inversa del resultado. Esto se debe a que es necesaria obtener la mínima distancia entre los sentidos, por ende se usa Dijkstra y también se sabe que dos sentidos están más relacionados mientras más cerca estén, entonces en el grafo se guardan los valores inversos como distancia. Es de esta forma que se construyó el grafo para cada uno de los textos (subjetivos u objetivos) existentes en el corpus.

1.4. Uso del PageRank

Una vez completo el grafo se requirió la ejecución del algoritmo de PageRank para determinar los sentidos adecuados de cada palabra existente en la oración que sea considerada en el grafo. Para la implementación del PageRank, se utilizó una adaptación descrita en Sobrevilla-Cabezudo et al. (2017) el cual es ejecutado en el grafo. Para la ejecución se requirió la siguiente información: (1) las aristas entre los vértices (puede tener uno o más sentidos de acuerdo a los enfoques vistos en la sección de construcción del grafo) de cada palabra que estén relacionados entre sí y (2) las frecuencias de uso que posee cada sentido la cual es obtenida a partir de la WordNet, esta información no se encuentra directamente en la WordNet por lo que se obtuvo usando el orden en que se encuentran los sentidos de cada palabra ya que la WordNet los ordena en base a cuales son los más usados frecuentemente.

El algoritmo de PageRank es descrito en la Figura 8

$$Pr = cMPr + (1 - c)v$$

Figura 8: Ecuación de PageRank

La ecuación del algoritmo cuenta con las siguientes variables, Pr el cual es el vector de PageRank donde se calcula el valor para cada vértice del grafo, es decir que si el grafo tiene "n" vértices el vector de PageRank será de tamaño "n". La variable "c" es una constante del algoritmo de PageRank llamada *damping factor*. La variable "M" es una matriz la cual depende de los vértices del grafo, es de tamaño "n" *x* "n", y su valor dependiendo del elemento es el siguiente: $M_{ji} = 1 / d_i$, siendo "j" la fila e "i" la columna donde se ubica el elemento de la matriz y el valor d_i representa la cantidad de aristas que salen desde el vértice i hacia algún otro vértice. Por último, el vector "v" es un vector de probabilidades de tamaño "n"*x*1, el cual es inicializado con el valor de 1/ "n" en cada elemento del vector.

Para la adaptación descrita por Sobrevilla-Cabezudo et al. (2017) se realizaron algunos cambios en algunas de las variables de la ecuación, las cuales son descritas a continuación: en la matriz M el valor descrito para M_{ji} cambia por la siguiente fórmula:

$$M_{ji} = \frac{w_{ij}}{\sum_{z} w_{iz}}$$

Figura 9: Nueva fórmula para la matriz usada por el algoritmo de PageRank

En esta fórmula (mostrada en la Figura 9) w_{ij} el cual es el numerador, representa el peso de la arista entre los vértices i y j, mientras que la sumatoria que es el denominador, es la suma resultado de todas las aristas que posee el vértice i. Además de este cambio el vector v, ya no es inicializado como 1/n en lugar de esto se usa el valor de frecuencia que se tiene de cada sentido (que está guardado en cada vértice) y a cada vértice le corresponde su respectivo valor de frecuencia entre la sumatoria total de frecuencias de todos los vértices del grafo.

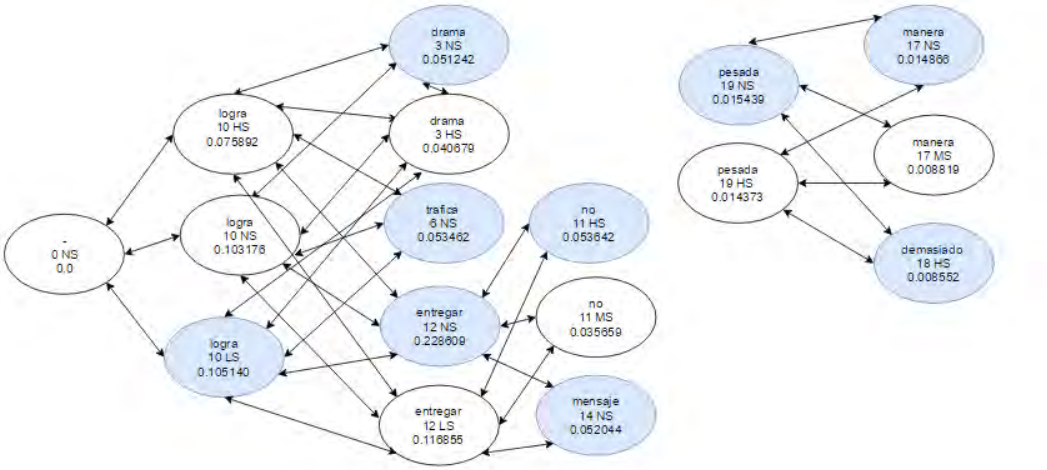

Figura 10: Grafo resultante de la oración con valores de PageRank

Tras la ejecución de este algoritmo se puede apreciar en la Figura 10 qué sentido o conjunto de sentidos son seleccionados como los más adecuados (esto para el caso de agrupar los sentidos por subjetividad) aunque la selección del sentido no es lo más importante sino el valor de subjetividad que posee dicho sentido o los conjuntos de sentidos seleccionados para el segundo caso, por lo que esa agrupación no genera ningún problema.

En la Figura 10 se puede ver que no están incluidas algunas palabras o expresiones como "mientras_que" debido a que las palabras que no se encuentren en la WordNet ni en la SentiWordNet no fueron consideradas. Por otra parte, hay otros vértices conectados entre sí en un subgrafo, esto es debido a que no hay conexión con el grafo principal, debido a que la palabra ("de") que conectaba esas palabras ("manera" y "entregar") fue descartada al no estar en la WordNet. Además de esto, para la ejecución del algoritmo de PageRank se tuvo en cuenta la siguiente consideración: solo se usaron los vértices que estén

conectado como mínimo a un vértice no vacío[11]. Entonces, las palabras consideradas para el algoritmo de PageRank se observan claramente en la Figura 10, siendo los sentidos seleccionados resaltados (con fondo azul, respecto a los demás que están con fondo blanco), en cada vértice del grafo está el valor obtenido de resultado por el algoritmo de PageRank, siendo asignado el valor subjetivo con mayor valor a la palabra correspondiente.

Concluida la ejecución y seleccionado el valor de subjetividad que corresponde a cada palabra que es vértice del grafo se procedió a generar las características que se utilizaron para el siguiente paso el cual es el algoritmo de clasificación. Dichas características serán más detalladas en el siguiente capítulo.

[11] (entenderse por vértice no vacío algún vértice que corresponda al sentido de una palabra, es decir no puede ser '-' ni '*')

2. Evaluación del algoritmo

En esta sección se analizará los resultados obtenidos por el algoritmo para determinar qué tan preciso fue en la detección de subjetividad de cada palabra o expresión presentes en el corpus. Para esta tarea se requieren los resultados obtenidos por el algoritmo de PageRank de la sección previa y los resultados de la anotación del corpus del capítulo anterior.

Los resultados obtenidos se presentan en la Tabla 5:

Algoritmos	Precisión	Exhaustividad	F1-Measure
Sentidos Juntos	76.84%	76.00%	76.36%
Sentidos Separados	77.47%	76.69%	77.02%
Sentido Más Frecuente	76.99%	76.20%	76.55%

Tabla 5: Tabla de Métricas (*Macro-Average*) de algoritmos a nivel de palabras

Estas medidas fueron obtenidas con respecto a las palabras que se utilizaron en los grafos mencionados en la sección previa de este capítulo. Se evaluaron las métricas descritas en los resultados obtenidos por los algoritmos usando para esto los datos obtenidos en la sección de anotación del corpus. Cabe resaltar que el énfasis del algoritmo utilizado para detección de subjetividad de palabras no es el de encontrar el sentido correcto de la palabra, sino encontrar el valor de subjetividad adecuado de la palabra, para saber que tanto influye sobre la subjetividad de todo el texto del que forme parte.

Los resultados muestran que los algoritmos están a la par, pero que el método que utiliza los sentidos agrupados en el grafo (sentidos separados) es el mejor de los propuestos.

Clasificación de oraciones objetivas y subjetivas

1. Implementación de los algoritmos

En esta sección se mostrarán los resultados de la experimentación con algunos algoritmos de clasificación para determinar si una oración es objetiva o subjetiva, a fin de determinar cuál de estos es el más apropiado para emplear junto con el algoritmo de detección. Los datos que recibirán estos algoritmos de clasificación serán las relaciones existentes entre los sentidos de los elementos de cada de oración, obtenidos del grafo generado previamente. Además de las relaciones también es importante la categoría que se le asigna a cada sentido siendo esta objetiva o subjetiva (alta, mediana y baja).

Terminada la ejecución del PageRank, se obtuvieron las características necesarias para el algoritmo de clasificación. Estas características se obtuvieron a partir de las aristas del grafo y la información que posee cada vértice (en este caso sería la categoría gramatical que posee la palabra y la categoría de subjetividad a la que corresponde, que fue determinada previamente). En la Figura 11 se puede observar, las características obtenidas como resultado del proceso descrito en el Capítulo 4.

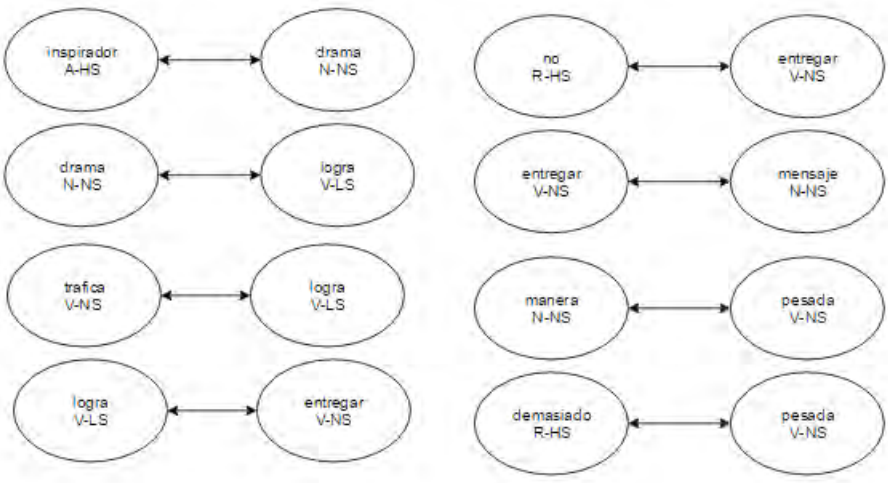

Figura 11: Características obtenidas del algoritmo de PageRank

Es con esta información que se generaron 136 características en su totalidad ya que las posibles categorías gramaticales son 4 (adjetivo, sustantivo, adverbio y verbo) mientras que las categorías de subjetividad también son 4 que fueron descritas previamente (no subjetivo, bajamente, medianamente y altamente subjetivo). Entonces cada característica corresponde a los vértices permitiéndoles ser de 16 diferentes tipos, por lo que las aristas al conectar cualquier tipo de vértice generan las características a utilizar, sin embargo algunas de las características estaban contadas más de una vez, como por ejemplo un verbo subjetivo unido a un sustantivo objetivo es igual que tener un sustantivo objetivo unido a un verbo subjetivo, sin embargo eran considerados como 2 características separadas, lo cual es incorrecto y por ende la cuenta final en lugar de ser de 256(16 tipos de datos que se relacionaban entre sí) se redujo a 136. A pesar de haber reducido las características, se consideró este que todavía eran demasiadas por lo que se utilizaron arboles aleatorios (*random decision tree*), para decidir qué características eran las mejores para la tarea de clasificación, reduciéndose así las características a 40.

Estas características se encuentran disponibles en el Anexo B.

2. Evaluación de resultados de los algoritmos

En esta sección se revisarán los resultados de los algoritmos de clasificación considerados para la aplicación, y se comparan los resultados y el desempeño con algunos algoritmos presentes en el estado del arte.

Los resultados mostrados en esta sección, se obtuvieron utilizando la data anotada con todo el procesamiento mostrado (ver en secciones previas) para obtener nuestros modelos entrenados. Luego de esto fue probado contra un porcentaje del corpus que estuvo sin anotar, como nuestra data de validación (10% o sea 500 oraciones con 250 objetivas y 250 subjetivas)

Estos resultados fueron realizados con distintos clasificadores y haciendo pruebas sobre distintos campos de estos los cuales se mostrarán en el Anexo C, además los criterios finales con los cuales se obtuvieron estos resultados se mostrarán en el Anexo D. Los resultados mostrados de todas las métricas son en valor promedio (*weighted-average*).

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	77%	74%	74%
Sentidos Separados	75%	73%	73%
Algoritmo sin Desambiguación de Sentido	68%	67%	67%

Tabla 6: Resultados de clasificación usando MLP.

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	79%	75%	76%
Sentidos Separados	79%	75%	76%
Algoritmo sin Desambiguación de Sentido	69%	69%	69%

Tabla 7: Resultados de clasificación usando SVM.

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	75%	74%	74%
Sentidos Separados	74%	72%	72%
Algoritmo sin Desambiguación de Sentido	68%	67%	67%

Tabla 8: Resultados de clasificación usando Regresión Logística.

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	77%	74%	75%
Sentidos Separados	76%	73%	74%
Algoritmo sin Desambiguación de Sentido	68%	68%	68%

Tabla 9: Resultados de clasificación usando Análisis discriminante lineal.

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	76%	68%	70%
Sentidos Separados	73%	64%	66%
Algoritmo sin Desambiguación de Sentido	66%	63%	64%

Tabla 10: Resultados de clasificación usando *K-Neighbors.*

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	70%	70%	70%
Sentidos Separados	67%	67%	67%
Algoritmo sin Desambiguación de Sentido	65%	62%	63%

Tabla 11: Resultados de clasificación usando arboles de decisión

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	84%	56%	63%
Sentidos Separados	81%	57%	63%
Algoritmo sin Desambiguación de Sentido	59%	59%	59%

Tabla 12: Resultados de clasificación usando bayesiano ingenuo.

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	76%	74%	74%
Sentidos Separados	75%	73%	74%
Algoritmo sin Desambiguación de Sentido	71%	70%	70%

Tabla 13: Resultados de clasificación usando gradiente descendente estocástica.

Algoritmos	Precisión	Exhaustividad	F-Measure
Sentidos Juntos	76%	75%	76%
Sentidos Separados	76%	75%	75%
Algoritmo sin Desambiguación de Sentido	70%	70%	70%

Tabla 14: Resultados de clasificación usando bosques aleatorios.

Analizando los resultados presentados (en las Tablas 6, 7, 8, 9, 10, 11, 12, 13 y 14), se puede ver que sí bien la clasificación de los textos usando el algoritmo desarrollado supera siempre al algoritmo base (que no utiliza desambiguación) además los resultados obtenidos son bastante adecuados en ambos modelos (grafos con sentidos juntos y separados) considerando todos los clasificadores probados. Con los resultados obtenidos es que se decide que el clasificador a emplearse en conjunto con el algoritmo de detección de subjetividad a nivel de palabras y/o expresiones, será el algoritmo *Support Vector Machine* (o SVM) para la aplicación final de detección de subjetividad, esto se debe a que obtuvo los mejores resultados considerando las 3 métricas empleadas. Cabe destacar que, si bien en la parte de detección a nivel de palabras los grafos con sentidos separados obtienen mejor resultados, en esta sección los grafos con sentidos juntos obtienen mejores resultados con cualquier método de clasificación. Por lo que asumiendo que la diferencia no es tanta en ambas secciones y que el resultado del clasificador final es muy importante se debería optar por trabajar con los grafos con sentidos juntos. Por otra parte, considerando que el modelo a usar es el SVM y que ambos grafos tuvieron resultados similares, también se puede optar por usar el modelo de grafos separados ya que a nivel de palabras obtuvo un mejor resultado. Entonces considerando lo mencionado, cualquiera de nuestros modelos implementados da buenos resultados, por lo que podría ser más adecuado dar más pruebas a fin de poder diferenciar cual es el mejor y/o anotar más datos para nuestros modelos.

Conclusiones y trabajos futuros

1. Conclusiones

En este trabajo de fin de carrera, se revisó distintos trabajos para entender los distintos posibles enfoques que intentan dar solución al problema de detección de subjetividad, para entender las carencias y los puntos que se pueden aprovechar. Cabe resaltar que estos trabajos tenían diversas limitaciones como el tipo de corpus que emplean o si consideraban o no la desambiguación de palabras.

Empezando con el trabajo, se tuvo que realizar la anotación del corpus con el que se probara el algoritmo para detección de subjetividad a implementar, como se ve en su sección (Corpus). Esto se debe a que el corpus no contenía información suficiente, para realizar las pruebas además de que no hay tantos corpus en español, por lo que se decidió trabajar con el propuesto y se necesitó completar la información que este presenta.

Luego de tener la información necesaria, para nuestro enfoque fue necesario determinar primero los valores de subjetividad de cada palabra (alto, medio, bajo o ninguno) de cada palabra perteneciente a un texto del corpus. Como se observa en la sección Subjetividad en palabras y/o expresiones, cada oración encontrada en el corpus pasa por un procesamiento, usando las distintas herramientas mencionadas, para poder generar un grafo que contenga la información necesaria para hacer uso del algoritmo de PageRank. Es con el grafo y el uso de dicho algoritmo que se logra asignar los sentidos adecuados a cada palabra y saber el respectivo valor subjetivo, además de conocer las relaciones existentes entre las palabras dentro del texto. La obtención de los sentidos será verificada con la información que proporciona el corpus, mientras que los demás resultados (valores y relaciones de las palabras) serán usadas para el siguiente paso.

Una vez obtenidos los resultados de la sección previa, se procede a clasificar los textos del corpus como objetivos o subjetivos, para esto como es descrito en la sección de clasificación de oraciones subjetivas y objetivas, se usaron características basadas en las relaciones de las palabras y su valor subjetivo. Estas características fueron usadas para entrenar clasificadores y ponerlos a prueba, para determinar cuál es el más adecuado para detectar si un texto es subjetivo u objetivo, para lograr esto se tuvo que dividir el corpus procesado y reducir las características con el fin de obtener el mejor resultado posible.

De acuerdo a los resultados obtenidos, si bien el método presenta un buen desempeño, se encuentra algunas complicaciones como por ejemplo algunas de las palabras o expresiones no fueron encontradas ni en la WordNet o SentiWordNet, por lo que no se pudieron incluir en el análisis y no se sabe que efectos hubiesen tenido en la clasificación de las oraciones. No solo esto, las categorías gramaticales identificadas para las palabras utilizando *Freeling* no siempre eran correctas, por lo que las características obtenidas para los clasificadores no eran las correctas. Sin embargo, en general podemos afirmar que los modelos implementados son muy buenos y que podrían brindar mejores resultados cubriendo algunos defectos mostrados.

2. Trabajos Futuros

Del alcance del trabajo y las conclusiones, se proponen los siguientes trabajos futuros:

- Probar el algoritmo con corpus de distintos idiomas, esto ayudaría a saber si se puede adaptar y utilizar en más contextos. Sería bastante útil comparar su desempeño con otras herramientas existentes en otros idiomas.
- Cubrir más tipos de opiniones presentes en el texto como el sarcasmo, ya que el algoritmo puede presentar errores en esos casos y al cubrirlos se puede mejorar el desempeño considerablemente.

- Probar el método con corpus de distinto dominio, como la información de Twitter (tweets), ya que la información está estructurada de una manera bastante distinta a la utilizada en este trabajo y sería útil probar si luego de adaptarse a dicho dominio, el desempeño obtenido resulta adecuado.
- Probar nuevas herramientas, para tener una mejor precisión en los resultados y no se afecte el desempeño de la aplicación final.
- Integrarlo con alguna herramienta para detección de polaridad y ver cómo cambia el desempeño de esta cuando use la detección de subjetividad.

Por último, se agrega que como resultado de esta investigación se logró escribir un artículo aceptado por la conferencia CICLING 2019 que ya se encuentra publicado, lo pueden buscar con el DOI 10.13053/CyS-23-3-3279 o en el anexo E.

Referencias

Baccianella, S., Esuli, A., & Sebastiani, F. (2010, May). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In LREC (Vol. 10, pp. 2200-2204).

Baur, A. W. (2016). Harnessing the social web to enhance insights into people's opinions in business, government and public administration. Information Systems Frontiers, 1-21.

Bazarova, N. N., Choi, Y. H., Schwanda Sosik, V., Cosley, D., & Whitlock, J. (2015, February). Social sharing of emotions on Facebook: Channel differences, satisfaction, and replies. In Proceedings of the 18th ACM conference on computer supported cooperative work & social computing (pp. 154-164). ACM.

Bouazizi, M., & Ohtsuki, T. (2016, May). Sentiment analysis: From binary to multi-class classification: A pattern-based approach for multi-class sentiment analysis in Twitter. In Communications (ICC), 2016 IEEE International Conference on(pp. 1-6). IEEE.

Brachman, R. J., & Anand, T. (1996). The process of knowledge discovery in databases. Advances in knowledge discovery and data mining.

Chaturvedi, I., Cambria, E., Zhu, F., Qiu, L., & Ng, W. K. (2015). Multilingual subjectivity detection using deep multiple kernel learning. In Proceedings of the Fourth International Workshop on Issues of Sentiment Discovery and Opinion Mining, KDD WISDOM.

Díaz, R. G. (2005). La lematización en español: una aplicación para la recuperación de información. Trea.

He, W., Tian, X., Chen, Y., & Chong, D. (2016). Actionable Social Media Competitive Analytics For Understanding Customer Experiences. Journal of Computer Information Systems, 56(2), 145-155.

Karimi, S., & Shakery, N. (2016) A language-model-based approach for subjectivity detection.

Khanna, S., & Shiwani, S. (2013) Subjectivity detection and Semantic orientation based Methods for Sentiment Analysis.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing systematic literature reviews in software engineering.

Liu, B. (2012). Sentiment analysis and opinion mining. Synthesis lectures on human language technologies, 5(1), 1-167.

Mihalcea, R.; Banea, C.; and Wiebe, J. (2007) Learning multilingual subjective language via cross-lingual projections, in Proceedings of the Association for Computational Linguistics (ACL), pp. 976–983, Praga, Czech Republic.

Miller, G. A. (1995). WordNet: A Lexical Database for English. Communications of the ACM 38, New York, NY, USA, pp. 39-41. ACM.

Narayanan, R., Liu, B., & Choudhary, A. (2009, August). Sentiment analysis of conditional sentences. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Proessing: Volume 1-Volume 1 (pp. 180-189). Association for Computational Linguistics.

Navigli, R. (2009). Word sense disambiguation: A survey. ACM Computing Surveys (CSUR), 41(2), 10.

Ortega, R., Fonseca, A., Gutiérrez, Y., & Montoyo, A. (2013). Improving subjectivity detection using unsupervised subjectivity word sense disambiguation. Procesamiento del lenguaje natural, 51, 179-186.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). The PageRank citation ranking: bringing order to the web.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

Petz, G., Karpowicz, M., Fürschuß, H., Auinger, A., Stříteský, V., & Holzinger, A. (2013). Opinion mining on the web 2.0–characteristics of user generated content and their impacts. In Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data (pp. 35-46). Springer Berlin Heidelberg.

Sixto, J., Almeida, A., & López de Ipiña, D. (2016) An Approach to Subjectivity Detection on Twitter Using the Structured Information.

Sobrevilla-Cabezudo, M. A., La-Serna-Palomino, N., & Maguina-Perez, R. (2015, October). Improving subjectivity detection for Spanish texts using subjectivity word sense disambiguation based on knowledge. Congreso Latinoamericano de Informática. IEEE.

Sobrevilla-Cabezudo, M. A., Oncevay-Marcos, A., & Melgar-Sasieta, H. A. (2017, *in press*) SenseDependency-Rank: A Word Sense Disambiguation Method Based on Random Walks and Dependency Trees.

Switzer, J., Khan, L., & Muhaya, F. B. (2011, August). Subjectivity classification and analysis of the ASRS corpus. In Information Reuse and Integration (IRI), 2011 IEEE International Conference on (pp. 160-165). IEEE.

Toutanova, K., Klein, D., Manning, C. D., & Singer, Y. (2003, May). Feature-rich part-of-speech tagging with a cyclic dependency network. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (pp. 173-180). Association for Computational Linguistics.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., & Martin, M. (2004). Learning subjective language. Computational linguistics, 30(3), 277-308.

Wiebe, J., & Riloff, E. (2005, February). Creating subjective and objective sentence classifiers from unannotated texts. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 486-497). Springer Berlin Heidelberg.

Anexos

Anexo A: Árbol de Problema

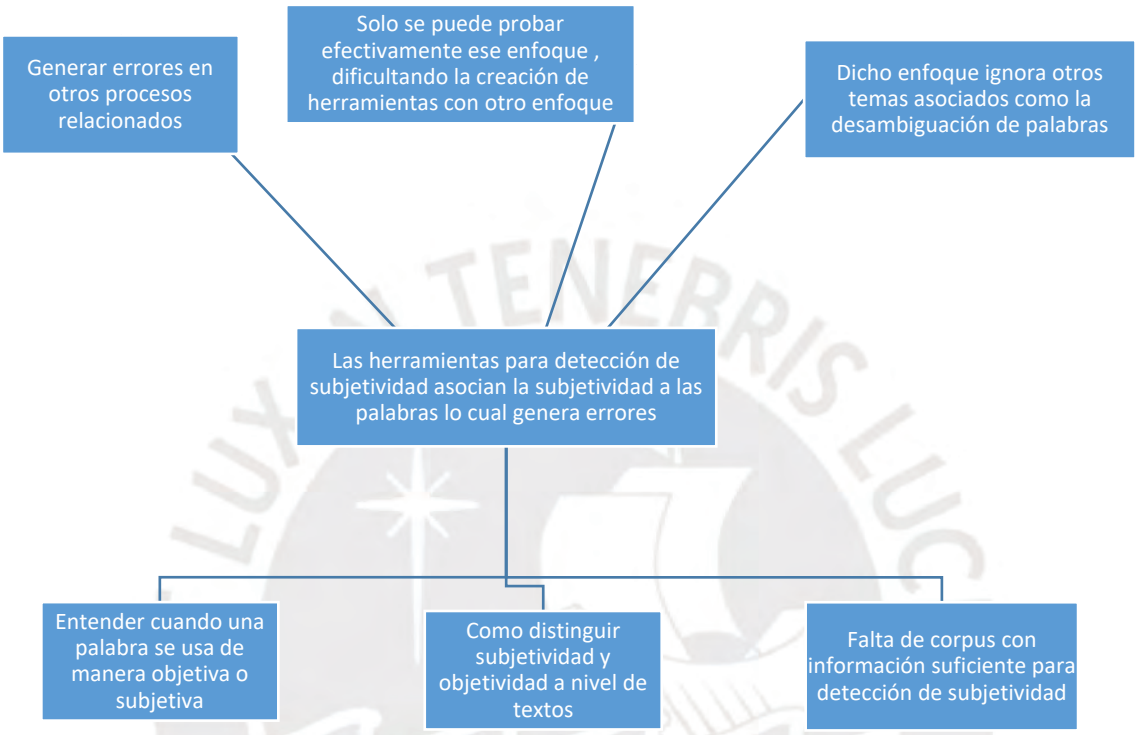

Figura Árbol de Problema

Anexo B: Lista de Características usadas en la Clasificación

Las características son extraídas a partir de las relaciones del *parser* y tienen el siguiente formato:

<Categoria$_1$>-<ValorSubjetivo$_1$> <Categoria$_2$>-<ValorSubjetivo$_2$>

donde las categorías gramaticales son sustantivo (N), verbo (V), adjetivo (A), adverbio (R); por sus escrituras en inglés y el valor subjetivo es NS (no subjetivo), LS (bajamente subjetivo), MS (medianamente subjetivo), HS (altamente subjetivo); por sus escrituras en inglés.

Los valores que asumen son numéricos que representan la cantidad de relaciones que siguen ese patrón.

Por ejemplo, en la Tabla 1 se puede ver la relación compuesta de un sustantivo altamente subjetivo y un adjetivo objetivo y la existencia de un patrón en la oración a analizar:

Característica	**Valor**
N-HS A-NS	1

Tabla Ejemplo de Característica

Así, fueron creadas 136 características de la combinatoria de todos los posibles casos. Después de un proceso de selección de características fueron seleccionadas 40 características que son mostradas en la Tabla 2.

V-NS V-MS	N-NS V-LS	A-NS V-HS	N-LS N-LS	N-NS R-MS
A-MS A-MS	V-LS V-HS	N-MS V-NS	R-LS V-LS	N-LS V-HS
R-NS V-MS	N-NS A-LS	R-MS R-HS	R-MS V-HS	R-MS R-MS
R-LS V-MS	N-MS V-HS	A-NS R-LS	R-NS R-MS	N-LS R-LS
N-MS R-NS	N-LS R-HS	R-HS V-HS	N-HS A-MS	N-LS A-HS
N-MS N-HS	R-NS R-LS	N-HS R-NS	A-NS V-NS	N-NS R-NS
A-NS A-LS	N-MS V-LS	A-LS R-LS	A-HS V-NS	N-HS V-NS
V-HS V-HS	A-NS R-NS	A-MS R-HS	A-HS A-HS	R-NS R-NS

Tabla Lista de Características usadas en el modelo de clasificación

Anexo C: Criterios de los Clasificadores (Grid-Search)

MLP:

'hidden_layer_sizes': [(12), (8,4,2) , (9,3)], ‘activation’: [‘logistic’ , ‘relu’ , ‘tanh’]

‘solve’: [‘adam’], alpha: [0.01,0.1,1,10,100], ‘random state’: [42],

‘learning_rate’: [‘constant’, ‘invscaling’, ‘adaptative’]

SVM:

'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000], 'kernel': ['linear','poly','rbf'],

'gamma' : 10.0**-np.arange(1,4)

Regresión Logística:

‘'C': [0.001, 0.01, 0.1, 1, 10, 100, 1000]

Análisis Discriminante Lineal:

'solver':['svd','lsqr','eigen']

K-Neighbors:

'n_neighbors':[1,3,5,7,9,11], 'weights':['uniform','distance'],

'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute']

Arboles de Decisión:

'criterion':['gini','entropy'], 'splitter':['random','best'], 'max_features':['sqrt','log2',None],

Bayesiano Ingenuo:

'priors':[None]

Gradiente Descendente Estocástica:

'loss':['hinge', 'log', 'modified_huber', 'squared_hinge', 'perceptron', 'squared_loss',

'huber','epsilon_insensitive','squared_epsilon_insensitive'],

'alpha':[0.0001,0.001,0.01,0.1,1], 'epsilon':[0.01,0.1,1]

Bosques Aleatorios:

'n_estimators': [5,10,15,20,25,30],

'criterion':['gini','entropy'],

'max_features': ['sqrt','log2', None]

Anexo D: Criterios Finales de los Clasificadores

MLP:

{'activation': 'logistic', 'alpha': 0.1, 'hidden_layer_sizes': 12, 'learning_rate': 'constant', 'random_state': 42, 'solver': 'adam'}

SVM:

{'C': 0.01, 'gamma': 0.001, 'kernel': 'linear', 'probability': True, 'random_state': 0}

Regresión Logística:

{'C': 1, 'random_state': 0}

Análisis Discriminante Lineal:

{'solver': 'lsqr'}

K-Neighbors:

{'algorithm': 'auto', 'n_neighbors': 5, 'weights': 'distance'}

Arboles de Decisión:

{'criterion': 'gini', 'max_features': None, 'random_state': 0, 'splitter': 'random'}

Bayesiano Ingenuo:

{'priors': None}

Gradiente Descendente Estocástica:

{'alpha': 0.1, 'epsilon': 0.01, 'loss': 'modified_huber', 'random_state': 0}

Bosques Aleatorios:

{'criterion': 'gini', 'max_features': 'log2', 'n_estimators': 30, 'random_state': 0}

Anexo E: Artículo Publicado

ISSN 2007-9737

An Exploratory Study of the Use of Senses, Syntax and Cross-Linguistic Information for Subjectivity Detection in Spanish

Rodrigo López, Daniel Peñaloza, Francisco Beingolea, Juanjose Tenorio, Marco Sobrevilla Cabezudo

Universidade de São Paulo,
Instituto de Ciências Matemáticas e de Computação,
Brazil

{a20112387, daniel.penaloza, francisco.beingolea, juanjose.tenorio}@pucp.pe, msobrevillac@usp.br

Abstract. This work presents an exploratory study of Subjectivity Detection for Spanish This study aims to evaluate the use of dependency relations, word senses and cross-linguistic information in Subjectivity Detection task. The first steps of this method include the labeling process of a Spanish corpus and a Word Sense Disambiguation algorithm. Then cross-linguistic English-Spanish information is obtained from Semcor corpus and used together with the Spanish data. Finally, this approach (using all gathered information and supervised algorithms) was tested showing better results than the baseline method in general.

Keywords. Subjectivity detection, dependency relations, wordnet, subjectivity word sense disambiguation, graphs, Spanish.

1 Introduction

Subjectivity Detection is the task that aims to determine whether a text is subjective or objective which means if it express and opinion or not [19]. According to [11], this task is considered to be more difficult than polarity classification; which focuses in recognize subjectivity as positive or negative. This could be due to different reasons such as non subjective sentences getting classified as positive or negative, an objective sentence implying an opinion which [8] describes as implicit opinion and more different cases.

Several studies on Sentiment Analysis have been performed, but most of them are in English including their tools and data [9], which is a reason to contribute with information from Spanish. Besides, classic methods attribute the subjectivity of text to the value of its respective words; ignoring some other factors such as their respective senses or the relations between them. According to [13], sentences may have ”sentiment words” but this is not enough to differentiate an opinion sentence from a non-opinion one. Considering the difficulties mentioned, some examples are shown below:

— *Mi teléfono se apaga una y otra vez*. (My cellphone turns off over and over again).

— *El nuevo Samsung Galaxy Note 7 es la bomba*. (The new Samsung Galaxy Note 7 is the bomb).

— *Había una bomba en la escuela*. (There was a bomb in school).

In the first example, an implicit opinion is shown, an objective sentence which express an opinion, in this case a negative one. About this sentence, it is significant to emphasize that the expression *una y otra vez* was associated with the word *apaga*, adding a new value to the sentence, making impossible to consider this sentence as an objective one. In the second example there is a subjective sentence with the word *bomba*, but the third one is an objective sentence with the same word, so a traditional classifier could have difficulties, since the senses of words are not considered.

This work presents an exploratory study of Subjectivity Detection for Spanish, which consider both the dependency relations of the words and word senses in the detection process.

Also, due to the lack of annotated resources (corpora with senses and subjectivity annotation) and in order to evaluate the cross-linguistic potential, we experimented the use of resources in English to train the subjectivity detector and compare their results with the training over a portion of an Spanish corpus manually annotated.

The paper is organized as follows, Related works are presented in Section 2. Section 3 discusses the work of gathering the knowledge for Subjectivity Detection. Section 4 is about testing the data obtained on Section 3 with supervised learning methods in order to see the subjectivity detection in sentences. Finally, Section 5 is about the final conclusions of this work.

2 Related Works

A semantic orientation-based approach is presented in [7]. Negation and POS-Tagging were used to choose the best features for subjectivity detection between uni-grams and phrases. SentiWordNet [1] was used with Point-wise Mutual Information (PMI) [4] to determine the semantic orientation of English documents, with good results using several features.

A study using information from Spanish tweets was proposed by [17]. Tweets include a lot of information besides the text which was exploited in this work. This information included unstructured and structured data. Thus, with these categories, different features were used such as emoticons, favorites, and retweets. After that, different supervised learning algorithms used each kind of data to get interesting results.

A framework for subjectivity detection using features in English and Spanish is shown in [3]. This framework uses Extreme Learning Machine(ELM), described in [6], but with Bayesian networks supporting its structure. Firstly, text is converted in a vector of words. This vector is processed in a deep convolutional neural network together with ELM. Finally, a Fuzzy Recurrent Neural Network is used to classify the initial text as positive, negative or neutral.

The work proposed by [14] presented an unsupervised Word Sense Disambiguation strategy for Subjectivity Detection in English. This approach relied on labeled data from different resources and information from SentiWordNet, since its focus was to just determine their subjectivity value. After this a rule-based method to classify sentences was used counting the words and testing it with supervised classifiers.

A rule-based method which uses knowledge from WordNet [12] and SentiWordNet for texts in Spanish is presented in [2]. It includes Word Sense Disambiguation using graphs with WordNet's senses to get subjectivity values. Then each word depending on its subjectivity value and its lexical category get a different weight to determine the subjectivity of a sentence. Besides, to get some of its parameters and values this work used the Semcor corpus in order to evaluate the usefulness of resources from other languages [10].

These studies have shown some important points such as: firstly, since there are not enough information from Spanish, using English resources should help with no problems, also the use of WordNet and SentiWordNet is really common. Secondly, there are different techniques but most of them rely on just the words, which means this work will be a good contribution. Thirdly, some approaches proved that using different features besides just the text may show great results. Finally, the use of Word Sense Disambiguation is helpful and improve the results for this task.

3 Subjectivity Detection

This work is based on graphs and dependency relations, which are used for a Word Sense Disambiguation method. Then, these results together with the same relations are used as features for supervised learning algorithms to determine the subjectivity of the sentences. The steps required for this, are explained in the next subsections.

3.1 Preliminaries

3.1.1 Corpus Annotation

In order to explore the subjectivity detection strategies for Spanish, the FilmAffinity corpus [2] was used. This corpus contains 2.500 objective

ISSN 2007-9737

sentences and 2.500 subjective sentences. In this paper, we will use the subjective sentence presented in Example 1 to explain the steps performed in our work.

Example 1. *Este inspirador drama, mientras que trafica con clichés, logra no entregar su mensaje de una manera demasiado pesada.*

Since this corpus only contains information about subjectivity for each sentence, and our work focused on exploring the use of fine-grained information in subjectivity detection, it was necessary to incorporate more knowledge into the corpus manually. In this case, information about senses were incorporated. Senses used were extracted from Multilingual Central Repository 3.0 (MCR) [5], which includes WordNets from different European languages (including Spanish) and is aligned with Princeton WordNet 3.0 [12] and SentiWordNet 3.0 [1] (which includes polarity information to senses).

The annotation process on content words, i. e., Nouns (N), Verbs (V), Adjectives (A) and Adverbs (R) used the MCR's senses[1]. To determine the words belonging to these grammatical categories and its respective lemmas, Freeling 4.0 [15] was used.

Due to annotation being a long and difficult task to be performed, just a small percentage of sentences of the corpus was annotated. This represented 8% (200 objective and 200 subjective sentences) of the corpus. The annotation was performed by four annotators with knowledge about Natural Language Processing. Besides the sense annotation, information from SUMO[2] was extracted, taking advantages from alignments between SUMO and MCR. SUMO contains information about some kind of attributes, specifically, any word may have more multiple attributes but in this case just *SubjectiveAssessmentAttribute* was extracted for the annotation.

Some annotated tokens of the Example 1 and its respective information are presented in Table 1. Column "Sense Identifier" contains the InterLingual Index. This is the WordNet's identifier and is useful to map between WordNets and SentiWordNet. In table 1, several senses may be seen for each word. For example, the word "drama" presents three senses associated with the synonyms "*dramaturgia* and *dramática*", "*obra teatral*", and "*evento dramático*" and "*tragedia*", respectively. Also, glosses and attributes from SUMO are presented for each word sense. With this information, annotators had to choose the sense more adequate in each sentence.

Table 1. Words and Senses Information

Word / Lemma	Tag	Sense Identifier	Synonyms	Gloss	Attributes
drama	N	spa-30-06376154-n	dramaturgia dramática	-	Text
		spa-30-07007945-n	obra_teatral	-	Text
		spa-30-07290278-n	evento_dramático tragedia	-	SubjectiveAssessment Attribute
demasiado	R	spa-30-00047392-r	excesivamente demasiadamente	-	SubjectiveAssessment Attribute
		spa-30-00415963-r	en_demasía excesivamente	más de_lo necesario	SubjectiveAssessment Attribute

3.1.2 Subjectivity Annotation

Even though senses were annotated, this study focused on subjectivity information, therefore, information from SentiWordNet was incorporated taking advantage of the alignments with MCR. SentiWordNet is focused on sentiment analysis and assigns positive, negative and neutral scores for each word sense, which must sum to 1. Thus, subjectivity score was defined as the sum of positive and negative scores and objectivity score was defined by the neutral score. After this, four subjectivity categories (non-subjectivity or NS, low subjectivity or LS, middle subjectivity or MS, and high subjectivity or HS) were defined according the subjectivity score. Table 2 presented the range of each category. Also, senses with *SubjectiveAssessmentAttribute* were annotated as HS.

Table 2. Subjectivity Categories

Category	NS	LS	MS	HS
Subjectivity Score Range	0	≤ 0.25	≤ 0.50	> 0.50

3.2 Subjectivity Word Sense Disambiguation (SWSD)

The first step in our proposal consisted in determining the subjectivity category for each word

[1] Available in `http://adimen.si.ehu.es/cgi-bin/wei/public/wei.consult.perl`

[2] Available in `http://www.adampease.org/OP/`

in a target sentence in order to use them to determine the subjectivity of a overall sentence. To achieve this step, an adaption to the work proposed by [18] was performed because our work was focused on disambiguating senses instead words.

3.2.1 Graph Building

Similar to [18], the graph for a sentence were built from its dependency tree. Thus, the nodes were defined by the senses of the words (obtained from MCR and SentiWordNet) and the edges were defined by the dependency relations included in the dependency tree.

Figure 1 shows the dependency tree of the Example 1 generated by Freeling, which will be used in this Section.

In order to evaluate the level of granularity of the nodes (in relation to senses and subjectivity), two configurations for the graphs were tested. The first one, called Separated Graph, considered a word-sense for each node. The second one, called Grouped Graph, considered a group of senses with the same subjectivity category for each node.

The weight of the edges was defined as the inverse of the distance between two nodes in the WordNet knowledge graph since it was considered that two senses are more related when they are closer in a graph. Distances were obtained from the application of Dijkstra algorithm on whole WordNet Knowledge Graph.

The weights in Separated Graphs were easier to be calculated (the definition before mentioned was used), since each node contained only one sense. In the case of grouped graphs, the weights were defined as the maximum value from the relations between the senses involved in each edge.

Figure 2 shows a subgraph of the graph generated for Example 1 considering the Grouped Graph configuration. As it may be seen, nodes contains one or more senses of a word according to the subjectivity category. For example, the node of the word "*logra*" belonging to the Low Subjectivity category (LS) groups four senses (sense identifiers are shown in Figure 2). Also, nodes are connected to other nodes according to their dependency relation. For example, the connection between "*logra*" and "*drama*" is defined by the dependency relation "*subj*" (it may be seen in Figure 1).

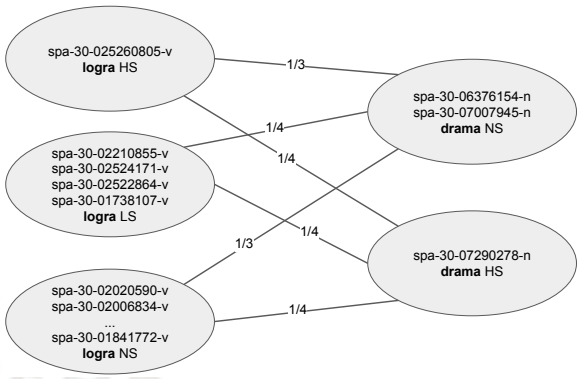

Fig. 2. Subgraph of the example sentence

3.2.2 SWSD

After Graph building, the subjectivity word sense disambiguation method was applied. Similar to [18], the PageRank algorithm [16] was executed. The Equation (1) shows the PageRank algorithm:

$$Pr = cMPr + (1 - c)v. \tag{1}$$

This equation is used in a graph (G) with N vertices, with these variables: The variable "Pr" will contain the result value for each vertex of the graph. The variable "c" is a constant from PageRank called damping factor. The variable "M" is an square matrix (NxN) with each element represented by the value $M_{ji} = 1/d_i$ if there is a relation between vertices "i" and "j", otherwise the value is 0. The value of d_i represents the number of edges going out from the i vertex. The variable "v" is a vector containing a value for each vertex of the graph and its value is usually $1/N$.

In this case, an adaptation of the algorithm used in [18] was used on both graph configurations. Thus, some changes were implemented. For example, the definition of cell values of matrix "M" was changed as shown in Equation (2) :

$$M_{ji} = \frac{w_{ij}}{\sum_z w_{iz}}. \tag{2}$$

ISSN 2007-9737

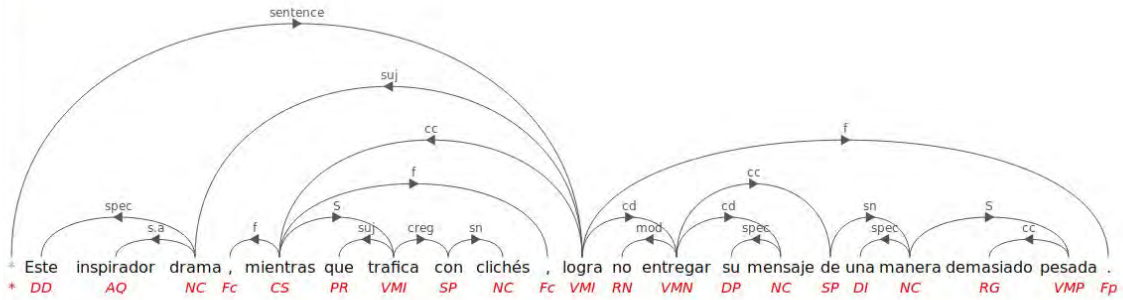

Fig. 1. Dependency Tree of Example 1

In this equation, w_{ij} is defined as the weight of the edge between vertices "i" and "j" and the sum considers all weights of edges that start from vertex "i". Besides that, the "Pr" vector was initialized with the value $1/N$ for each vertex and vector "v" had the value of frequency obtained from the WordNet as probabilities for each vertex. Finally, the damping factor used was 0.85 and the number of iterations was 30.

3.3 Bringing Cross-linguistic Knowledge

Since the already described corpus annotation is an important but laborious task, gather more data was necessary. Considering there were not many resources for a non-English language, cross-linguistic knowledge was used. Besides, in Section 2 was shown that using English resources could be useful and it was an opportunity to evaluate their influence in the results.

In this case, the Semcor corpus was used. Semcor contains 20.138 sentences annotated with WordNet's senses but it is not tagged with subjectivity classification at sentence-level. This way, OpinionFinder 2.0[3] [20] was executed to label which sentences were objective and subjective automatically. OpinionFinder[4] is a sentiment analysis tool, which shows a precision (91.7%) in the Subjectivity Detection task. After the execution of OpinionFinder, 934 objective sentences and 934 subjective sentences were selected to compose the English corpus.

[3]Available in `http://mpqa.cs.pitt.edu/opinionfinder/opinionfinder_2/`

[4]This tool was used because shows good results in the Subjectivity Detection task.

To conclude with this section, it is important to note that since the tools used for this corpus were different, there were some information that were not available like it was for FilmAffinity's corpus. For example, SUMO ontology was not available neither the relations between senses from WordNet, due to labels differences; only the senses and its subjectivity values were found. Also, since there was a lot of sentences in the Semcor to check subjectivity manually, the OpinionFinder was used. So, these differences between how both corpus were worked may lead to different results, that will be describe later.

3.4 Incorporating Syntactic Knowledge

In order to evaluate the contribution of syntactic knowledge, several experiments were performed. Firstly, the words and their subjectivity were considered as features, since the majority of works rely on this. So, 16 features were considered, due to 4 subjectivity categories and 4 grammatical categories, being called grammatical features.

Secondly, with the obtained relations, the proposal of this work was to use them together with the words and categories as features. Mixing together the previous 16 grammatical features with each other according to their relations, resulting in 136 features called dependency features. Next, it was decided to mix both the the grammatical and dependency features to evaluate the their use. Finally, all these features were used with different supervised learning methods. Some examples of the dependency features, using Example 1, are shown in Table 3.

ISSN 2007-9737

Table 3. Final Features

Relations	Features
inspirador - drama	A-HS N-NS
drama - logra	N-NS V-LS
demasiado - pesada	R-HS V-NS

4 Results and Discussion

4.1 Corpus Annotation

In relation to the corpus annotation, it is important to regard the following details:

— We used the senses from WordNet with information in Spanish, however sometimes that was not enough to choose the appropriate sense, so information from WordNet in English was checked too and when there was not enough information to resolve the confusion with similar senses, the most frequent one was consider the right one.

— There were cases when an appropriate sense could not be found for a word in both Spanish and English; so, a lemma that could be considered as a similar one, in the specific context, was used to search the right sense.

— There were words that were classified wrongly by the POS-tagger, so in those cases, the POS-tag was changed for an adequate one and the search used the lemma with its new tag, with the first and/or second consideration if necessary.

— When the first, second and third items were not enough for a word, it was left blank, since there were cases when a word has no meaning by itself (such as proper names or modal verbs) making it impossible to find any sense at all.

Finally, 4.620 words (belonging to 400 sentences extracted of the original corpus) were annotated and used to evaluate the recall of the subjectivity word sense disambiguation method (SWSD).

4.2 Subjectivity Word Sense Disambiguation

In relation to the SWSD, the results obtained from the SWSD method (graphs grouped and separated) were compared with a baseline. In our case, The Most Frequent Sense (MFS) was selected as baseline. This heuristic works in the following manner: All words were labeled with its most frequent sense from the WordNet. The results are shown in Table 4.

Table 4. SWSD summary

	Graphs					MFS		
	Group	R	Sep	R	Total		R	Total
Noun	1210	0.82	1199	0.81	1473	1817	0.83	2198
Verb	686	0.63	716	0.66	1084	750	0.65	1162
Adj.	625	0.74	623	0.74	840	665	0.74	897
Adv.	266	0.84	269	0.85	316	303	0.83	363
	2787	0.75	2807	0.76	3713	3535	0.77	4620

Table 4 shows the results of all methods tested. As it may be seen, using Separated Graphs produced better results than Grouped Graphs, even though this difference could be not significant. Also, the results of SWSD using separated graphs outperformed the results for MFS in all grammatical categories, except for nouns, producing a worse, although not significant, overall performance (due the frequency of annotated nouns).

These results may be explained by different reasons. For example, there were problems with the tools used, as explained earlier in this section, and a small amount of data could be annotated. Besides, our method could not analyze all the data, since we use relations between words, but that was not the case for MFS.

Finally, it is important to mention that the most common mistakes in the algorithms used for WSD in this work happened with verbs a significant number of times. This could be explained by all the problems already mentioned in the annotation process; such as problem with the tools (POS-tagger) or finding the appropriate sense for a word, since some words, specially verbs, are associated with a big number of senses making it more difficult for the algorithm.

ISSN 2007-9737

4.3 Subjectivity Detection

As mentioned in Section 3.3 and Section 3.4, we evaluated the use of Semcor (English corpus) and the use of dependency relations in subjectivity detection task. Also, we tested the usefulness of subjectivity word sense disambiguation in subjectivity detection. Thus, we experimented the following configurations: grouped graphs and separated graphs; training on Semcor, FilmAffinity and both corpus together; and training using dependency features, grammatical features and both features together.

All experiments were performed using Linear SVM algorithm (C value was 0.01) with all features normalized (without feature selection or dimensionality reduction). Also, a non-swsd baseline was used. Specifically, this baseline do not use WSD to obtain the subjectivity from words, but this is defined by the mean score of all its respective senses. Besides, we compared our results with the proposal presented in [2]. This method used an rule-based method and a subjectivity word sense disambiguation algorithm to perform subjectivity detection in the same corpus. In order to evaluate our experiments, we tested on a sub-corpus of the FilmAffinity corpus. This corpus was composed by 500 sentences (250 objective and 250 subjective). Besides, to evaluate the use of general features, another method was compared with our proposal, which used Bag of Words (BOW) and TF-IDF together with the FilmAffinity corpus.

Table 5 shows the results of all experiments performed. We may say that using SWSD and our methods specially grouped graphs show a slightly better performance than the baseline in all the experiments, except for the FilmAffinity with grammatical features which showed the best results. This may be related with nouns which have more presence in the corpus and most of them tend to be N-NS or N-HS with objective and subjective sentences respectively, according to the labeled corpus. Besides, noun senses may be from any subjectivity category, so the graph methods may be making more mistakes than taking the mean score of the senses. Then, comparing to the other works (BOW and [2]), all our best methods (training of FilmAffinity and using grammatical features) outperformed its results, being BOW which got the worst results. One point to highlight is that work proposed in [2] used Semcor as training corpus and obtained results comparable with methods which used the same features and the same corpus.

In relation to the cross-linguistic knowledge, FilmAffinity corpus showed the best and most consistent results for all the features and methods, which showed that the Semcor corpus may not be compatible with this work. Specifically, some subjective texts in FilmAffinity corpus were composed by 2 or 4 sentences together, unlike the Semcor, where it never happened. Then, since there is a relatively difference between the size of both corpus, it was evident that Semcor had a lot more senses and dependency relations. So, considering the differences described between tools, corpus data, words and/or features; it does not seem like both corpus could be used together or that good results would come out from using the English information.

Finally, dependency features were useless in all experiments, even harming the performance when mixing with grammatical features. One possible reason is that Freeling still suffers dealing with dependency relations. Thus, we could lose lot of information from a sentence, leading to worse results.

In order to perform a deep analysis, we analyzed false positives, with some points to remark. In models with Semcor corpus most of the errors were related to features with the category HS, since other categories were dominant the presence of this could be confused easily by the classifiers. Next, with the FilmAffinity corpus, the mistakes were related specifically with the most common features from 2 categories, being A-HS and N-NS this association was really common, so it was easily confused. However, this happened in specific situations like sentences with few relations including this feature or when using words and relations together, since the words have more weight due to being more increasing the probability of errors.

After this, with both corpus together the mistakes were similar due to FilmAffinity corpus being small

Table 5. Subjectivity Detection Results

Method	Training Corpus	Features	Objectivity			Subjectivity			
			P	R	F1	P	R	F1	Average F1
Grouped Graphs	Semcor	Dependency	0.76	0.64	0.70	0.58	0.71	0.64	0.67
		Grammatical	0.58	0.78	0.67	0.84	0.67	0.74	0.71
		Dependency + Grammatical	0.66	0.69	0.67	0.71	0.67	0.69	0.68
	FilmAffinity	Dependency	0.89	0.70	0.78	0.62	0.85	0.71	0.76
		Grammatical	0.88	0.75	0.81	0.71	0.85	0.77	**0.79**
		Dependency + Grammatical	0.88	0.72	0.79	0.66	0.84	0.74	0.77
	Semcor + FilmAffinity	Dependency	0.76	0.75	0.76	0.74	0.76	0.75	0.75
		Grammatical	0.71	0.79	0.75	0.81	0.74	0.77	0.76
		Dependency + Grammatical	0.74	0.76	0.75	0.76	0.74	0.75	0.75
Separated Graphs	Semcor	Dependency	0.75	0.64	0.69	0.59	0.70	0.64	0.67
		Grammatical	0.56	0.78	0.65	0.84	0.66	0.74	0.71
		Dependency + Grammatical	0.64	0.68	0.66	0.70	0.66	0.68	0.67
	FilmAffinity	Dependency	0.88	0.70	0.78	0.63	0.84	0.72	0.76
		Grammatical	0.86	0.74	0.79	0.70	0.83	0.76	**0.78**
		Dependency + Grammatical	0.85	0.71	0.78	0.66	0.81	0.73	0.76
	Semcor + FilmAffinity	Dependency	0.77	0.73	0.75	0.71	0.76	0.73	0.74
		Grammatical	0.68	0.78	0.73	0.81	0.72	0.76	0.75
		Dependency + Grammatical	0.72	0.75	0.74	0.76	0.73	0.74	0.74
Baseline	Semcor	Dependency	0.40	0.68	0.51	0.82	0.58	0.68	0.63
		Grammatical	0.28	0.90	0.42	0.97	0.57	0.72	0.67
		Dependency + Grammatical	0.36	0.74	0.48	0.88	0.58	0.70	0.64
	FilmAffinity	Dependency	0.71	0.69	0.70	0.68	0.70	0.69	0.69
		Grammatical	0.77	0.81	0.79	0.82	0.78	0.80	**0.80**
		Dependency + Grammatical	0.68	0.75	0.71	0.78	0.71	0.74	0.73
	Semcor + FilmAffinity	Dependency	0.53	0.72	0.61	0.79	0.63	0.70	0.67
		Grammatical	0.49	0.85	0.62	0.91	0.64	0.75	0.71
		Dependency + Grammatical	0.58	0.75	0.65	0.81	0.66	0.73	0.70
[2]	Semcor	Grammatical	0.74	0.60	0.66	0.66	0.78	0.72	0.70
BOW	FilmAffinity	TF-IDF	0.00	0.00	0.00	1.00	0.50	0.67	0.67

in comparison, but it is interesting to note that this mix of corpus improved the results from Semcor. As a final point, it is important to mention that Semcor was checked (around 400 sentences) and a lot of mistakes were found, since most of the sentences looked like objective ones, which could be due to tools used or to the kind of text from the corpus, since it is related to news. The sentences were corrected, but with a small positive change in the results, so it confirmed that the used of Semcor was not the best for this work.

5 Conclusions and Final Remarks

In this paper, an exploratory study about subjectivity detection for Spanish was presented. We explored the use of Word Sense Disambiguation to identify senses' subjectivity; the incorporation of syntactic information to subjectivity detection; and the use of cross-linguistic information, specifically English, to train supervised models for Subjectivity Detection.

The SWSD was on pair with the selected baseline, so considering that the results were not exactly bad, gathering more labeled data will we important to the evaluation of this method in order to see how the results might change in all parts of this work. Then, before considering the subjectivity detection of texts, the Semcor corpus was used for the experiments.

Considering differences in tools, data, knowledge and with the final results it was determined that the information from English was not compatible with this work, or that maybe Semcor was not an appropriate corpus due to its nature or being labeled inaccurately either by its senses

or by the OpinionFinder. Finally the experiments proposed here showed good results for the subjectivity detection task for both kind of graphs, with grouped graphs being better, proving that this approach is useful and other works will benefit from it.

Finally, some future works are related to annotate more data from FilmAffinity, to see if the results may be improved; testing data from another domain in order to see if the results change; using appropriate data from English or another language, labeling the information if necessary; and finally to use some of these features with a polarity classification tool to evaluate its usefulness.

References

1. **Baccianella, S., Esuli, A., & Sebastiani, F. (2010).** Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. *LREC*, volume 10, pp. 2200–2204.
2. **Cabezudo, M. A. S., Palomino, N. L. S., & Perez, R. M. (2015).** Improving subjectivity detection for Spanish texts using subjectivity word sense disambiguation based on knowledge. *Latin American Computing Conference (CLEI)*, IEEE, pp. 1–7.
3. **Chaturvedi, I., Ragusa, E., Gastaldo, P., Zunino, R., & Cambria, E. (2018).** Bayesian network based extreme learning machine for subjectivity detection. *Journal of The Franklin Institute*, Vol. 355, No. 4, pp. 1780–1797.
4. **Church, K. W. & Hanks, P. (1990).** Word association norms, mutual information, and lexicography. *Computational linguistics*, Vol. 16, No. 1, pp. 22–29.
5. **Gonzalez-Agirre, A., Laparra, E., & Rigau, G. (2012).** Multilingual central repository version 3.0. *LREC*, pp. 2525–2529.
6. **Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006).** Extreme learning machine: theory and applications. *Neurocomputing*, Vol. 70, No. 1-3, pp. 489–501.
7. **Khanna, S. & Shiwani, S. (2013).** Subjectivity detection and semantic orientation based methods for sentiment analysis. *International Journal of Scientific and Engineering Research*.
8. **Liu, B. (2012).** Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, Vol. 5, No. 1, pp. 1–167.
9. **Lo, S. L., Cambria, E., Chiong, R., & Cornforth, D. (2017).** Multilingual sentiment analysis: from formal to informal and scarce resource languages. *Artificial Intelligence Review*, Vol. 48, No. 4, pp. 499–527.
10. **Mihalcea, R. (1998).** Semcor semantically tagged corpus. *Unpublished manuscript*.
11. **Mihalcea, R., Banea, C., & Wiebe, J. (2007).** Learning multilingual subjective language via cross-lingual projections. *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 976–983.
12. **Miller, G. A. (1995).** WordNet: a lexical database for English. *Communications of the ACM*, Vol. 38, No. 11, pp. 39–41.
13. **Narayanan, R., Liu, B., & Choudhary, A. (2009).** Sentiment analysis of conditional sentences. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, Association for Computational Linguistics, pp. 180–189.
14. **Ortega, R., Fonseca, A., Gutiérrez, Y., & Montoyo, A. (2013).** Improving subjectivity detection using unsupervised subjectivity word sense disambiguation. *Procesamiento del Lenguaje Natural*, Vol. 51, pp. 179–186.
15. **Padró, L. & Stanilovsky, E. (2012).** Freeling 3.0: Towards wider multilinguality. *LREC2012*.
16. **Page, L., Brin, S., Motwani, R., & Winograd, T. (1999).** The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.
17. **Sixto, J., Almeida, A., & López-de Ipiña, D. (2016).** An approach to subjectivity detection on Twitter using the structured information. *International Conference on Computational Collective Intelligence*, Springer, pp. 121–130.
18. **Sobrevilla-Cabezudo, M. A., Oncevay-Marcos, A., & Melgar, A. (2017).** Sense dependency-rank: A word sense disambiguation method based on random walks and dependency trees. *International Conference on Computational Linguistics and Intelligent Text Processing*, Springer, pp. 185–194.
19. **Wiebe, J. & Riloff, E. (2005).** Creating subjective and objective sentence classifiers from unannotated texts. *International conference on intelligent text processing and computational linguistics*, Springer, pp. 486–497.
20. **Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., & Patwardhan, S. (2005).** Opinionfinder:

ISSN 2007-9737

A system for subjectivity analysis. *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pp. 34–35.

Article received on 14/02/2019; accepted on 04/03/2019.
Corresponding author is Rodrigo López.