Semi-supervised learning of types of temporal meanings in the Spanish lexicon

Pablo Grill
Facultad de Ingeniería (UdelaR)
Montevideo, Uruguay
Email: pablogrill21@gmail.com

Mathias Claassen
Facultad de Ingeniería (UdelaR)
Montevideo, Uruguay
Email: mathiasclaassen7@gmail.com

Aiala Rosá
Facultad de Ingeniería (UdelaR)
Montevideo, Uruguay
Email: aialar@fing.edu.uy

Hernán Correa
Facultad de Humanidades y
Ciencias de la Educación (UdelaR)
Montevideo, Uruguay
Email: hcorreasuarez@gmail.com

***Resumen*—This paper presents a series of semi-supervised learning algorithms which were designed to classify words or expressions with temporal meanings. The algorithms use a set of pre-tagged temporal expressions and a set of semantic classes which were defined within a research project on the lexical coding of temporal meaning in Spanish. The algorithms in this article are mostly based on word embeddings, but they also make use of other methods. The results obtained strongly depend on the temporal classes considered, but, for some classes, results have reached 90 % precision or above.**

***Index Terms*—Temporal networks, Word embeddings, Semi-supervised learning, Semantic classification, Natural Language Processing.**

I. Introducción

El Proyecto de Redes Temporales[1], definido por el Departamento de Teoría del Lenguaje y Lingüística General de la Facultad de Humanidades y Ciencias de la Educación de la UdelaR (DTLLG), se enfoca en la asignación de rasgos a expresiones temporales. Se define como expresión temporal a todas aquellas unidades (palabras o frases) que contengan en alguno de sus significados un rasgo de temporalidad. Por ejemplo, las expresiones *almorzar, anciano, vez, antes, ahora, de repente, inmediato, minuto, a las apuradas, permanecer, durante* y *por los siglos de los siglos* pueden considerarse como temporales. Considerando que en dicha definición se menciona "alguno de sus significados"; una expresión puede ser considerada como temporal aunque su acepción más habitual no lo sea. En este proyecto se parte de 15 rasgos temporales definidos en base a criterios semánticos. Considerando que estos criterios no son los más habituales cuando nos referimos a temporalidad, las redes temporales son un enfoque innovador del tema para el área.

Los rasgos definidos son: *Anclaje, Deíctico, Desplazamiento, Duración, Fase, Frecuencia, Individuo, +/- Delimitado, +/- Especificado, +/- Recurrente, Orden, Puntual, Tiempo-Espacio, Tiempo-Manera* y *Transformatividad*. La definición semántica de los mismos se detalla en el anexo VI.

Al analizar las definición de los rasgos se aprecia que son heterogéneos; cubriendo diferentes características de la temporalidad en el idioma español. Por ejemplo, comparemos los rasgos *Orden* y *Duración*:

- Orden: *Expresiones que se caracterizan porque sus denotados mantienen relaciones de orden. Ejemplos: Edad Media, agosto, siguiente, viernes.*
- Duración: *Expresiones que designan propiedades exclusivamente temporales y las duraciones que se asocian a ellos. Ejemplos: sempiterno, duradero, continuar, durar, el curso del tiempo, por los siglos de los siglos.*

La única relación que existe entre ambos rasgos es la presencia de la temporalidad, en los demás aspectos definen dos conjuntos distintos.

Otra particularidad de los rasgos temporales son aquellos que se definen de forma "+/-". En estos casos, el rasgo se define mediante dos subconjuntos disjuntos. Por ejemplo, el rasgo *+/- Recurrente* se define de la siguiente manera: *Expresiones que designan estadios recurrentes o no recurrentes. Las expresiones Más Recurrentes se repiten cíclicamente, como por ejemplo, las estaciones, los días de la semana o los nombres de eventualidades culturales. Ejemplos: Semana de Turismo, verano, mañana, abril, viernes. Por otro lado, las expresiones Menos Recurrentes no se repiten. Ejemplos: infancia, Neolítico, clásico, pasado.*

Como se puede apreciar, los conjuntos *Más Recurrente* y *Menos Recurrente* están relacionados al rasgo recurrencia, pero son semánticamente diferentes. Por lo tanto, podrían considerarse cada uno de ellos como rasgos independientes.

Otro aspecto interesante de la red temporal es que cada expresión puede pertenecer a más de un rasgo. Por ejemplo, la expresión *Abril* (cuarto mes del año) pertenece a los rasgos *+/-Recurrente, Orden, +/-Delimitado* y *+/-Especificado*. Otra consideración que se debe tener es la forma en que la red considera las palabras ambiguas. En caso de que dos definiciones de una palabra puedan considerarse temporales, la clasificación de cada definición puede ser diferente. Por ejemplo, la palabra Abril tiene las siguientes definiciones temporales:

- Cuarto mes del año. Tiene 30 días.

- Primera juventud.

Respecto al rasgo *+/-Recurrente*, la primera definición se corresponde al conjunto *Más Recurrente* y la segunda al conjunto *Menos Recurrente*.

Bajo las características anteriormente expuestas, desde el punto de vista del Procesamiento de Lenguaje Natural (PLN), la clasificación de expresiones temporales se puede considerar como un problema de clasificación semántica *multiclass* (múltiple categoría) y *multilabel* (cada expresión puede pertenecer a más de un rasgo). En este caso, el problema de clasificación cuenta con las siguientes características particulares:

- *Bajo volumen de datos ya clasificados*. Se dispone de aproximadamente 15 ejemplos por cada rasgo. Este valor es demasiado pequeño para abordar el problema con métodos de clasificación estándares.
- *Similitud de rasgos (límites poco claros)*: Algunos de los rasgos que se definen son muy similares, diferenciandose entre ellos por particularidades semánticas muy especificas. Diferenciar estos rasgos de forma automática es un desafío. Sumado a esto, hay rasgos que son subconjuntos de otros rasgos.
- *Definición de rasgos muy específicos*: Algunos de las rasgos definidos son tan específicos que determinar si una expresión temporal debe pertenecer a ellos genera discusión entre los propios integrantes del equipo de investigación del DTLLG.
- *Conjunto de expresiones temporales en construcción*: El conjunto de expresiones temporales no está del todo definido ya que los criterios que definen la temporalidad pueden ir variando a medida que se profundice en la clasificación de las palabras. Para este trabajo se consideró un universo de aproximadamente 1500 palabras temporales, definidas manualmente por integrantes del DTLLG.
- *Conjunto de rasgos en construcción*: Los 15 rasgos definidos pueden llegar a variar una vez que se tenga un conjunto mayor de expresiones clasificadas.
- *Redes Temporales innovadoras*: No se dispone de trabajos previos sobre el tema de forma que se pueda comparar los resultados.

Debido a estas características, muchos de los abordajes comunes a los problemas de clasificación no pudieron ser utilizados y fue necesario elaborar nuevos algoritmos que se ajustarán al contexto de las redes temporales. En este trabajo se aborda el problema de la asignación de rasgos temporales mediante un enfoque basado principalmente en la aplicación de representaciones vectoriales. Se toma como hipótesis de trabajo que palabras con semántica similar compartirán los mismos rasgos de la red temporal. Sumado a las representaciones vectoriales, se complementa el enfoque con otras técnicas como lo son la *Similitud de Lesk*[2] y la incorporación de categorías gramaticales.

El resto del artículo se organiza en cinco secciones. En la sección *Trabajos relacionados* se mencionan otros trabajos que utilizan modelos vectoriales para tareas de clasificación semántica. En la sección *Algoritmos desarrollados* se detallan los diferentes algoritmos que se elaboraron, mostrando los resultados obtenidos en la sección *Resultados obtenidos*. En la sección *Conclusiones* se mencionan las conclusiones que se pudieron extraer del trabajo realizado. Por último, se agrega una sección *Anexo* en la que se detalla las definiciones de cada rasgo de la red temporal.

II. Trabajos relacionados

Debido al enfoque innovador de las redes temporales, no se dispone de trabajos previos sobre el tema. Considerando que uno de nuestros objetivos es estudiar la aplicación de modelos vectoriales para este problema particular, se estudiaron algunos trabajos en donde se aplican modelos vectoriales para tareas de clasificación.

Los modelos vectoriales de palabras (*Word Embeddings* [3], [4], [5]) son muy utilizados en la actualidad para abordar problemas de clasificación semántica debido a los buenos resultados que obtienen. En líneas generales, un modelo vectorial de palabras consiste en la codificación de las palabras y/o frases de un idioma en un vector numérico de grandes dimensiones (de 100 a 500). Esta codificación permite tener un mapeo de "(palabra, vector)" que permite identificar cada palabra con su correspondiente vector. Este método de codificación de palabras posee algunas características que lo hacen más interesante de utilizar respecto a otros métodos:

- *Generación automática*: Existen algoritmos de generación automática de modelos vectoriales a partir de corpus de texto.
- *Vectores densos de grandes dimensiones*: Ideales para utilizar en redes neuronales.
- *Contenido semántico en los vectores*: Los vectores se generan de forma que su representación sea fiel a la semántica de la palabra.

A causa de estas propiedades y características, los modelos vectoriales pueden ser utilizados para analizar palabras y las relaciones entre ellas. Se pueden definir distancias y funciones algebraicas en el modelo vectorial que pueden mapearse a relaciones gramaticales o semánticas entre las palabras asociadas a dichos vectores.

Al analizar la utilización de los modelos vectoriales para la resolución de problemas de clasificación semántica se encuentran diversos artículos que tratan el tema. A continuación citaremos algunos en los que basamos nuestro trabajo.

En [6] se comparan varios algoritmos de generación de modelos vectoriales respecto a Word2Vec[3]. Además, el artículo se encarga de definir y explicar claramente los hiperparámetros que acepta Word2Vec para la generación del modelo vectorial y como su variación modifica los resultados obtenidos. Dichos hiperparámetros se pueden clasificar en *Pre-processing hyperparameters*, *Association Metric Hyperparameters* y *Post-processing Hyperparameters*. Comprender la utilización de los hiperparámetros fue fundamental para la tarea de generación de modelo vectorial.

Una de las utilizaciones más usuales de los modelos vectoriales es para tareas de análisis de opiniones o sentimientos.

El análisis de opiniones/sentimientos se basa fuertemente en el contenido semántico de una oración. Si bien la polaridad de una opinión no es igual a la clasificación temporal, ambos problemas radican en el contenido semántico de una expresión.

En [7] se utilizan vectores de palabras para realizar análisis de opiniones en twitter. En este caso, se genera un modelo vectorial especifico a partir de métodos supervisados partiendo de recursos léxicos de opinión. Otro ejemplo de análisis de sentimiento con modelos vectoriales se presenta en [8]. En este trabajo se utiliza un modelo vectorial generado de forma no supervisada que es refinado de forma que se represente mejor la polaridad de las expresiones. En este artículo se afirma que los métodos automáticos de generación suelen generar buenos modelos vectoriales para las tareas de *Word Sense disambiguation*, *Named Entity Recognition*, *PoS Tagging* y *Document Retrieval*. En [9] se utilizan diversos modelos vectoriales para realizar análisis de sentimiento en Twitter en el marco de la competencia *SemEval 2016 Task4: Sentiment Analysis in Twitter*. En este trabajo se utilizan tres modelos vectoriales diferentes, uno de ellos generado de forma genérica y los otros dos orientados al análisis de sentimientos.

De todos estos trabajos, se puede apreciar que los modelos vectoriales pueden generar buenos resultados en tareas de clasificación semántica. Sin embargo, todos los trabajos debieron refinar o contruir modelos específicos orientados a su contexto (análisis de sentimiento) para obtener mejores resultados.

Por último, el artículo [10] fue muy importante para el desarrollo del presente trabajo ya que define la *similitud coseno*. La similitud coseno se define matemáticamente mediante la fórmula 1:

$$cosine\_similarity(A, B) = \frac{\langle A, B \rangle}{\|A\| * \|B\|} \quad (1)$$

A partir de la definición de similitud coseno se define la distancia coseno con la fórmula 2:

$$cosine\_distance(A, B) = 1 - cosine\_similarity(A, B) \quad (2)$$

En este artículo, se muestra que la similitud coseno es una buena representación de la similitud semántica de las palabras. Es decir, aquellas palabras cuya similitud coseno sea mayor serán más similares que aquellas palabras cuya similitud coseno sea menor.

III. Algoritmos desarrollados

El trabajo realizado se enfocó en el diseño, desarrollo e implementación de algoritmos que permitieran la clasificación automática de nuevas expresiones temporales según los rasgos.

Los algoritmos se basan en modelar los rasgos temporales a partir de un conjunto de palabras. Este enfoque hace que todos los rasgos se traten de igual manera, otorgando escalabilidad y flexibilidad a los algoritmos. Considerando que los rasgos pueden ser modificados en etapas posteriores del proyecto, contar con algoritmos de estas características es un factor fundamental.

Considerando que la clasificación de las expresiones depende de su definición, la red temporal necesita asociar el par *(palabra, definición)* a cada uno de los rasgos. Entre los algoritmos que desarrollamos algunos contemplan esta particularidad y otros no (consideran todas las definiciones de la palabra de igual manera).

Otra característica de los algoritmos es que se idearon para una utilización semisupervisada. Se tomó esta decisión ya que los casos clasificados originales eran demasiado pocos para realizar una clasificación supervisada. En este contexto, los algoritmos se ejecutan de la siguiente manera:

- Se parte de un conjunto inicial de datos ya clasificados para cada rasgo de la red temporal (aproximadamente 15 palabras por rasgo).
- Se ejecuta alguno de los algoritmos desarrollados.
- El algoritmo devuelve para cada rasgo de la red un conjunto de expresiones candidatas a tener el rasgo.
- Un integrante del DTLLG clasifica manualmente cada una de las expresiones candidatas.
- La información ingresada manualmente se incorpora a la base de conocimientos de los algoritmos de forma de disponer de mayor información en la siguiente iteración.

Esta estrucutra de algoritmo se asemeja a los algoritmos basados en Active Learning [11] con la particularidad de que en cada iteración se devuelve la totalidad de los candidatos y no solo un conjunto seleccionado de los mismos. De esta forma, mediante la ejecución iterativa de los algoritmos se va incrementado el conjunto de datos clasificados de la red temporal permitiendo alcanzar en cada iteración resultados más precisos. Sumado a esto, los algoritmos contienen una serie de parámetros que permiten ir modificando sus comportamientos de forma de poder adecuarlos al conjunto de palabras ya clasificadas.

Todos los algoritmos se basan en la premisa de que expresiones de “semántica similar” contendrán los mismos rasgos semánticos en la red temporal. Bajo esta premisa, los algoritmos desarrollados se enfocan fuertemente en encontrar expresiones "similares semánticamente“.

Los insumos de que disponen los algoritmos son los siguientes:

- Conjunto de rasgos temporales con ejemplos (inicialmente unos 15 ejemplos por rasgo).
- Conjunto de expresiones temporales a clasificar junto a sus definiciones.
- Modelo vectorial de palabras y frases del idioma español. En nuestras pruebas, utilizamos un modelo vectorial de dimensión 500 generado con Word2Vec a partir del corpus SBWCE[12].
- Biblioteca Gensim para el manejo del modelo vectorial[13].
- Clase gramatical de las expresiones. En nuestras pruebas, la generamos utilizando la herramienta Freeling[14].

Todos los algoritmos desarrollados se encuentran disponibles en una herramienta web que permite su ejecución y la clasificación de resultados. De esta forma, se facilita al usuario

del DTLLG la ejecución semisupervisada de los algoritmos. Además, se permite visualizar los datos de la red temporal, ya sea desde el punto de vista de las expresiones como desde el punto de vista de los rasgos. Esta herramienta esta siendo utilizada por el DTLLG para profundizar en la construcción de la red temporal. Sin embargo, no se descarta que un futuro pueda utilizarse para otros problemas de clasificación semántica.

Los algoritmos desarrollados se describen en las subsecciones siguientes.

III-A. Coarse Tags clusters

Algoritmo que se basa fuertemente en la Similitud Coseno y el modelo vectorial de palabras para la clasificación. Las expresiones sugeridas serán aquellas que se encuentren cerca (mediante la similitud coseno) de las expresiones pertenecientes al rasgo.

El algoritmo puede resumirse en los siguientes pasos:

- Para cada rasgo, separar las expresiones pertenecientes a él en base a su clase gramatical (Coarse Tag de Freeling).
- Por cada clase gramatical (en el contexto de un rasgo), en función de las expresiones de dicha clase, crear un vector representante de las mismas.
- Por cada expresión no clasificada, realizar los siguientes pasos:
  - Identificar la clase gramatical de la expresión.
  - Por cada rasgo, realizar los siguientes pasos:
    - Buscar el vector representante del rasgo dado para la clase gramatical de la expresión. Si el rasgo no tiene un vector para dicha clase, la palabra no pertenece al rasgo.
    - Evaluar la similitud coseno del vector representante respecto al vector de la expresión.
    - Si la similitud coseno se encuentra dentro de un umbral definido, la expresión se considera como candidata. En caso contrario, se descarta.

El umbral de similitud utilizado por el algoritmo es parametrizable por el usuario, tomando del valor de 0.45 por defecto. Cuanto mayor sea el valor del umbral, más restrictivos serán los resultados propuestos. La sugerencia es utilizar el valor 0.45 en las primeras etapas de clasificación de forma de generar muchos candidatos para agregar a los rasgos. A medida que se van agregando más palabras a las rasgos, se puede ir aumentando el valor del umbral de forma de obtener menos resultados, pero más precisos.

El vector representante de cada clase gramatical se determina en base a un conjunto de palabras positivas y palabras negativas. El algoritmo de generación del vector permite que este último se encuentre "cercano" a las palabras positivas, pero "lejano" a las palabras negativas utilizando para ello la similitud coseno. Es importante que el número de palabras negativas sea menor que el de palabras positivas para que el vector generado sea representativo. El conjunto de palabras positivas y negativas puede ser seleccionado por el usuario de forma que se modele el rasgo de mejor manera. El algoritmo de generación de dicho vector se basa en la implementación de la función *most_similar* de la biblioteca Gensim. En dicha función se genera internamente un vector que cumple lo requerido para el vector representante de una clase gramatical.

Los resultados obtenidos por este algoritmo son muy buenos para aquellos rasgos que cuentan con un núcleo bien definido, es decir, todas las expresiones pertenecen al rasgo se encuentran "cerca" en el espacio vectorial.

Este algoritmo es especialmente útil en las primeras etapas para generar un conjunto de ejemplos clasificados más grande, ya que no necesita demasiados ejemplos clasificados para proponer buenos candidatos.

Una desventaja de este algoritmo es que no maneja la ambigüedad; es decir, todas las definiciones de las expresiones se agrupan de igual manera en la red temporal. Esto sucede ya que la clasificación se determina en base al vector asociado, el cual agrupa todos los usos de la palabra. Otra desventaja de este algoritmo es que requiere que la expresión temporal tenga un vector asociado. En el caso de las frases, no todas tienen un vector asociado lo que provoca que no puedan ser clasificadas por este algoritmo.

III-B. Distancia a N

Este algoritmo es una alternativa al *Coarse Tags Cluster*. El algoritmo propone expresiones que están a una distancia dada de al menos *N* expresiones de un rasgo. Esto permite obtener buenos resultados para aquellos rasgos que no tienen un núcleo bien definido en el espacio vectorial, sino que tienen una disposición más bien dispersa en varios grupos.

La estructura del algoritmo puede definirse de la siguiente manera:

- Por cada expresión no clasificada, realizar los siguientes pasos:
  - Por cada rasgo de la red:
    - Evaluar la similitud coseno de la expresión respecto a cada expresión representante del rasgo.
    - Si el número de expresiones que se encuentran dentro de un umbral es superior a N, la palabra se considera como candidato del rasgo. En caso contrario la palabra no se considera como candidato.

Este algoritmo permite parametrizar los valores del umbral de similitud y el de *N*. Al igual que el algoritmo de coarse tag clusters, al aumentar el umbral de similitud se obtendrán valores más restrictivos, pero más precisos. El valor de *N* dependerá de cada rasgo, ya que está muy asociado a su dispersión. Se recomienda utilizar valores bajos en las primeras etapas de forma de generar muchos candidatos y refinarlos en las siguientes iteraciones.

Considerando que este algoritmo también se basa en el modelo vectorial cuenta con las mismas desventajas que *Coarse Tag Clusters*.

III-C. Expansión por sinónimos

Este algoritmo no usa el modelo vectorial, sino que se basa exclusivamente en las definiciones de las expresiones

utilizando la similitud de Lesk.

El algoritmo de Lesk fue publicado por Michael Lesk en 1986 y se usa principalmente para desambiguar sentidos de palabras (Word Sense Disambiguation). Esto es determinar qué sentido semántico es utilizado en un contexto específico. Otro uso del algoritmo es la detección de palabras similares en base a sus definiciones. En este caso, se comparan las definiciones de dos palabras y en base a la cantidad de palabras en común se determina su similitud. Esta utilización del algoritmo se denomina *Similitud de Lesk*.

El algoritmo *Expansión por sinónimos* crea un grafo de similitud a partir del algoritmo de Lesk y algunas reglas de similitud predefinidas manualmente. Utilizando dicho grafo, se proponen expresiones que son "similares" a expresiones que pertenecen al rasgo. Esta similitud se define en base a cuántos pasos tiene el camino más corto entre una expresión que pertenece al rasgo y la expresión objetivo.

El algoritmo cuenta con dos parámetros: el primero es la profundidad hasta la cual se le permite navegar en el grafo de similitudes. Si este valor es -1, entonces se considerará el grafo completo. Si el camino entre dos palabras es mayor que este valor, entonces no se toma como válido.

El segundo parámetro permite especificar el porcentaje de palabras que deben ser "similares" a una palabra para considerarla candidata. Dicho de otro modo, si un rasgo tiene 20 palabras positivas y el valor de este parámetro es 0.1 (10 %), para que una palabra sea candidata, deberá ser similar a al menos 2 palabras. Esto pretende evitar la inclusión de palabras que solamente son similares a una sola palabra. Variando el valor de este parámetro se pueden obtener candidatos más o menos seguros.

Este algoritmo es muy útil en todas las etapas, pues no requiere muchos ejemplos iniciales y va a sugerir palabras nuevas aun cuando haya muchas palabras en el rasgo. Se recomienda especialmente ejecutarlo entre ejecuciones de otros algoritmos para "expandir" el conjunto de palabras clasificadas. Cabe destacar que este algoritmo generará mejores resultados si las definiciones provistas de las palabras son más específicas y detalladas.

A diferencia de los algoritmos anteriores, en este caso, las diferentes acepciones de una misma expresión pueden ser clasificadas de forma independiente ya que la clasificación se basa en la definición de la expresión.

III-D. Sugerencias de palabras usando "most similar" de Gensim

A los algoritmos de clasificación se agregó la funcionalidad de sugerir nuevas expresiones temporales. Para ello utiliza la función *most_similar* que provee Gensim sugiriendo las palabras del modelo vectorial que están más cerca de las palabras positivas de cada rasgo.

El parámetro que se puede usar para este algoritmo es cuántas palabras se desea que se sugieran. El propósito de este algoritmo es sugerir nuevas palabras temporales que no fueran consideradas por el DTLLG en una primera instancia.

IV. Resultados obtenidos

De los algoritmos detallados anteriormente, en este artículo se evalúa de forma cuantitativa el algoritmo de *Coarse Tags clusters*. Los resultados detallados corresponden a una primera iteración del algoritmo. Es decir, son el resultado de partir de aproximadamente 15 palabras ya clasificadas por rasgo. Las mediciones de desempeño se realizan rasgo a rasgo. Se toma esta decisión ya que los rasgos son heterogéneos e independientes; por lo tanto, el problema de asignación de cada rasgo puede verse como un problema de clasificación independiente, generando resultados diferentes.

Considerando los pocos ejemplos con los que se contaba para trabajar, no se pudo separar un conjunto de datos de testing para realizar evaluaciones. Por lo tanto, algunas de las métricas estándar no se pueden utilizar. Las métricas que se utilizaron para la evaluación de los resultados son las siguientes:

- Precisión de las expresiones predichas: El DTLLG evaluó el conjunto de expresiones predichas de forma de medir la precisión alcanzada en cada rasgo.
- Falsos negativos: Aplicando el algoritmo a los propios integrantes del rasgo.
- Total de expresiones predichas: Como no se puede medir el *recall* se mide la cantidad total de predicciones por rasgo.

Al analizar de forma global los resultados obtenidos se aprecia que los mismos son muy buenos, superando las expectativas iniciales en algunos de los rasgos. Desde un punto de vista cuantitativo, los algoritmos desarrollados demuestran ser efectivos para los rasgos *Anclaje*, *Orden*, *+Delimitado*, *Frecuencia* e *Individuo*. En estos rasgos, el número de falsos positivos en las expresiones que se predicen es bastante bajo. Esto permite que la asignación de estos rasgos de forma automática pueda realizarse con la certeza de que la calidad de dicha asignación será buena. Además, el número de predicciones es bastante elevado, lo que permite aumentar en pocas iteraciones el número de expresiones asociadas a dicho rasgo. En el caso de los otros rasgos, la precisión obtenida con los algoritmos no es suficientemente elevada como para permitir la asignación del rasgo de forma automática. En estos casos, es necesaria una revisión humana previa a la asignación del rasgo. Los resultados de aplicar el algoritmo *Coarse Tag Cluster* en la primer iteración se pueden apreciar en el cuadro I. En las siguientes iteraciones, utilizando los diferentes algoritmos, se fueron mejorando los resultados. Esto sucede ya que se dispone de una mayor cantidad de palabras clasificadas y por ende mayor información para cada rasgo. De esta forma, iteración a iteración se logra mejorar la precisión y aumentar el número de palabras predichas.

Si analizamos los resultados desde un punto de vista cualitativo, los resultados obtenidos también presentan un balance positivo. Si bien los resultados numéricos muestran un correcto desempeño para 5 rasgos de los 18, la utilización que el DTLLG le dará a los algoritmos desarrollados permite obtener valor de todos las rasgos. En los rasgos en que la precisión

Categoría	Cantidad de Palabras Categoría	Cantidad de Predicciones	Falsos Positivos	Precisión
$+Delimitado$	37	113	7	0,94
$Individuo$	28	41	4	0,90
$Anclaje$	33	65	10	0,84
$Frecuencia$	23	29	5	0,83
$Orden$	34	87	17	0,80
$Deictico$	20	40	14	0,65
$+Especificado$	21	79	30	0,62
$-Especificado$	31	74	28	0,62
$Tiempo-Manera$	24	24	10	0,58
$Tiempo-Espacio$	14	19	8	0,57
$+Recurrente$	16	72	32	0,55
$Duracion$	22	42	19	0,55
$Puntual$	26	99	60	0,39
$-Recurrente$	12	62	38	0,38
$Desplazamiento$	19	26	16	0,38
$Fase$	33	89	58	0,35
$Transformatividad$	35	47	39	0,17
$-Delimitado$	16	50	46	0,08

Cuadro I
RESULTADOS OBTENIDOS

está en el entorno de 0.4 y 0.6, se podrán obtener resultados productivos en una utilización semisupervisada de los algoritmos. Los rasgos que presentaron una precisión menor también servirán al DTLLG ya que se les brindará un universo de palabras menor en el que enfocar la clasificación manual (solo las palabras predichas), lo que agilizará el trabajo del equipo de investigación.

Si analizamos los algoritmos desarrollados, podemos ver que los resultados que arrojan dependen del contexto en que se encuentren. Podríamos resumir las asociaciones contexto-algoritmo de la siguiente forma:

- **Coarse Tags clusters**: Algoritmo útil para realizar la expansión de rasgos cuando se dispone de pocos elementos clasificados. Esto se puede lograr, ya que gran parte de la información que utiliza el algoritmo proviene del modelo vectorial. Este algoritmo es útil cuando se dispone de palabras que son utilizadas comúnmente y, por lo tanto, para las que son modeladas de buena forma por los vectores asociados a ellas. Además, es necesario que la definición asociada a la palabra con la que se está tratando sea la de uso más frecuente, de modo que el vector obtenido del modelo se pueda asociar a la definición.
- **Distancia a N**: Considerando que este algoritmo se basa también fuertemente en el modelo vectorial, los contextos de utilización son iguales que los del algoritmo Coarse Tags clusters.
- **Expansión por sinónimos**: Este algoritmo no utiliza el modelo vectorial, sino que se basa fuertemente en la definición de las expresiones. Este algoritmo es útil cuando las expresiones temporales con las que se trabaja contienen definiciones completas. Usar las definiciones de las expresiones permite desambiguar los significados de las expresiones y asignar rasgos distintos a cada uno de ellos.

Considerando la variedad de algoritmos y sus diferentes contextos, podemos ver que el trabajo realizado es bastante completo ya que permite cubrir diferentes escenarios.

V. CONCLUSIONES

Podemos concluir que el balance del trabajo realizado es positivo. Para algunos rasgos, se logró obtener buenos resultados mediante clasificación automática. Considerando los escasos ejemplos ya clasificados con los que se contaba, alcanzar estos valores fue un hito muy importante.

Un aspecto fundamental de este trabajo fue la utilización del modelo vectorial de palabras. Este insumo permitió agregar información semántica muy importante para los algoritmos de clasificación desarrollados. Como resultado del trabajo se obtiene una utilización de los modelos vectoriales para un problema de clasificación semántica específico.

Otro aspecto del trabajo que nos parece positivo destacar fue la utilización de la similitud de Lesk para un problema que no es el usual. El algoritmo de similitud de Lesk se utiliza usualmente para la desambiguación de palabras. En este caso se utilizó para realizar clasificación semántica.

Además, el trabajo provee a los integrantes del DTLLG información sobre la red temporal que deberán analizar. El modelo vectorial sugirió candidatos a expresiones temporales que no habían sido considerados en una primera instancia (por ejemplo, algunos períodos religiosos). Por otra parte, los resultados de los experimentos permitieron detectar rasgos que presentaban mejores resultados que otros. El hecho de que la clasificación automática para algunos rasgos presente resultados tan malos podría indicar que la definición del mismo es demasiado específica o amplia y que ésta debería revisarse. Sumado a esto, los investigadores del DTLLG cuentan con una herramienta web que les permite ejecutar los diferentes algoritmos con el fin de avanzar en la estructura y composición de la red temporal.

En líneas generales, considerando los buenos valores obtenidos en los rasgos *Anclaje*, *Orden*, *+Delimitado*, *Frecuencia*

e *Individuo* y la variedad de algoritmos desarrollados que cubren diversos contextos, podemos concluir que los resultados obtenidos son muy buenos.

VI. Anexo

Definición de rasgos:

- **Anclaje**: Expresiones relativas que se anclan en alguna eventualidad. El concepto de eventualidad se aplica a procesos, eventos y a los resultados de estos. Por ejemplo: *hasta, desde, tan pronto como, ni bien*.
- **Deíctico**: Expresiones deícticas que remiten al momento de la enunciación. Ejemplo son: *hoy, ahora, de ahora en mas, entonces*.
- **Desplazamiento**: Expresiones relativas que designan desplazamientos en relación con un momento o tiempo de referencia. Ejemplos: *alargar, dilatar, adelantado, precoz, precipitarse*.
- **Duración**: Expresiones que designan propiedades exclusivamente temporales y las duraciones que se asocian a ellos. Por ejemplo: *sempiterno, duradero, continuar, durar, el curso del tiempo, por los siglos de los siglos*.
- **Fase**: Expresiones que designan entidades cuyos significados se definen en relación con fases temporales. Ejemplos son: *albores, comienzos, cesar, juventud, todavía*.
- **Frecuencia**: Expresiones que designan frecuencia de algún hecho o evento. Por ejemplo: *cotidiano, frecuente, intermitente, de vez en cuando*.
- **Individuo**: Expresiones que designan individuos definidos en relación con su ciclo vital. Ejemplos: *adolescente, cordero, osezno, ternera, cincuentón*.
- **+/- Delimitado**: Expresiones que designan extensiones temporales delimitadas, o bien, no delimitadas, es decir, que poseen un inicio y un final o que no lo tienen. Ejemplos del subconjunto **Más Delimitado** son: *fin de semana, siglo, infancia, racha, para largo*. Por otro lado, ejemplos de **Menos Delimitado** son: *eterno, infinito, nunca, jamás de los jamases, inmortal*.
- **+/- Especificado**: Expresiones que en conjunto designan extensiones caracterizadas por el hecho de que empiezan y culminan, es decir, delimitadas. Por lo tanto son una subcategoría del rasgo *Más delimitado*. Se distinguen dos subconjuntos en relación con la información que hace referencia a los límites. El subconjunto **Más Especificado** contiene piezas cuyos significados contienen una información precisa acerca del punto de inicio y de culminación, ambos intrínsecamente establecidos. Ejemplos de este subconjunto son: *mes semestre, fin de semana, Renacimiento, feriado.* Por otro lado, el subconjunto **Menos Especificado** contiene piezas cuyos significados carecen de esta información. Ejemplos son: *pasajero, transitorio, rato, temporada, a fines de, ocaso*.
- **+/- Recurrente**: Expresiones que designan estadios recurrentes o no recurrentes. Expresiones **Más Recurrentes** se repiten cíclicamente, como por ejemplo estaciones, días de la semana o nombres de eventualidades culturales. Ejemplos: *Semana de Turismo, verano, mañana, abril, viernes.* Por otro lado, las expresiones **Menos Recurrentes** no se repiten. Ejemplos son:*infancia, Neolítico, clásico, pasado*.
- **Orden**: Expresiones que se caracterizan porque sus denotados mantienen relaciones de orden. Ejemplos: *Edad Media, agosto, siguiente, viernes*.
- **Puntual**: Expresiones que designan puntos temporales para los que idealmente su comienzo y final coinciden en el tiempo. Por ejemplo: *instante, medianoche, repentino, estirar la pata, nacimiento*.
- **Tiempo-Espacio**: Expresiones que designan relaciones espacio-temporales. Ejemplo: *aceleración, lento, celeridad, como bala*.
- **Tiempo-Manera**: Expresiones en las que coexisten informaciones de tiempo y de manera. Por ejemplo: *inesperadamente, lentamente, precipitadamente, raudo, a todo trapo*.
- **Transformatividad**: Expresiones que denotan cambios de estado o cambios en un proceso. Ejemplos son: *nacer, finalizar, envejecer, volverse, evolución*.

Agradecimientos

Agradecemos a todos los integrantes del Departamento de Teoría del Lenguaje y Lingüística General de la Facultad de Humanidades y Ciencias de la Educación (DTLLG) por la disponibilidad que demostraron durante todo el proyecto.

Referencias

[1] "La codificación de la temporalidad en el léxico del español," in *Investigaciones actuales en Lingüística. Vol II: Morfología y Lexicología.* Alcalá de Henares: Servicio de Publicaciones de la Universidad de Alcalá.

[2] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone," in *Proceedings of the 5th Annual International Conference on Systems Documentation*, ser. SIGDOC '86. New York, NY, USA: ACM, 1986, pp. 24–26.

[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 01 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[4] TensorFlow, "Vector representation of words," https://www.tensorflow.org/tutorials/word2vec/, accedido: 08-02-2017.

[5] Google, "Word2vec training," https://code.google.com/archive/p/word2vec/, accedido: 01-09-2016.

[6] O. Levy, Y. Goldberg, and I. Dagan, "Improving distributional similarity with lessons learned from word embeddings," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 211–225, March 2015.

[7] E. M. Cámara, M. A. G. Cumbreras, M. T. M. Valdivia, and L. A. U. López, "SINAI-EMMA: Vectores de palabras para el análisis de opiniones en twitter," Universidad de Jaén, Tech. Rep., Septiembre 2015.

[8] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 142–150. [Online]. Available: http://dl.acm.org/citation.cfm?id=2002472.2002491

[9] Y. Zhou, Z. Zhang, and M. Lan, "Ecnu at semeval-2016 task 4: An empirical investigation of traditional nlp features and word embedding features for sentence-level and topic-level sentiment analysis in twitter," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 256–261. [Online]. Available: http://www.aclweb.org/anthology/S16-1040

[10] O. Levy and Y. Goldberg, “Linguistic regularities in sparse and explicit word representations.” in *CoNLL*, R. Morante and W. tau Yih, Eds. ACL, 2014, pp. 171–180. [Online]. Available: http://dblp.uni-trier.de/db/conf/conll/conll2014.html
[11] S. Arora and S. Agarwal, “Active learning for natural language processing,” Literature Review, Carnegie Mellon University, 2008.
[12] C. Cardellino, “Spanish Billion Words Corpus and Embeddings,” March 2016. [Online]. Available: http://crscardellino.me/SBWCE/
[13] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.
[14] L. Padró and E. Stanilovsky, “Freeling 3.0: Towards wider multilinguality,” in *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*. Istanbul, Turkey: ELRA, May 2012.