Tratamiento de Expresiones Temporales
con Redes Neuronales Artificiales y Representaciones Distribuidas de las Palabras

Mathias Etcheverry

Instituto de Computación
Universidad de la República Oriental del Uruguay
Pedeciba Informática

Orientadora: Dina Wonsever

Tesis de Maestría

Septiembre 2016

Agradecimientos

Quiero agradecer a mi orientadora Dina Wonsever por su apoyo a lo largo de este trabajo, por hacerme numerosas correcciones y sugerencias, por introducirme al procesamiento del lenguaje natural y por recibirme en el grupo de PLN. Quiero agradecer al equipo docente que llevó durante muchos años el curso aprendizaje automático: Guillermo Moncecchi, Diego Garat y Raúl Garreta, porque gracias a este curso recibí por primera vez el concepto de una red neuronal artificial.

Quiero agradecer a todos los integrantes del grupo de PLN por formarme y soportarme. A Pablo Ezzati por brindarme un recurso informático con una unidad de procesamiento gráfico, esto me permitió ejecutar gran parte de los experimentos presentados en tiempos viables. A Alejandro Martínez y Agustín Azzinnari porque en la elaboración del proyecto que realizaron construyeron el repertorio de vectores de mejor calidad que conozco para el español.

Quiero agradecer a la ANII por el apoyo económico, al programa SticAmSud por el apoyo para la pasantía en París y a la UdelaR y al Pedeciba por el apoyo para los congresos LKE2015 y LREC2016.

Por último y más importante, quiero agradecer a mi familia, a mi novia (quien leyó este informe reiteradas veces realizando muchísimas correcciones) y a mis amigos. Gracias por ser una fuente recurrente de energía, este trabajo no habría sido posible sin ellos y es a quiénes está dedicado.

Resumen

En esta tesis se realiza el reconocimiento y la clasificación de expresiones temporales en español sin incluir otra información explícita del dominio que los datos de entrenamiento. El enfoque propuesto consiste en modelos de redes neuronales artificiales que toman como entrada representaciones vectoriales de las palabras. Estas representaciones están construidas en base a la distribución de los contextos en los que ocurren y los modelos son entrenados utilizando textos anotados con la información temporal que se pretende aprender.

Por un lado, se estudia si las representaciones vectoriales, construidas de forma no supervisada, junto con los modelos neuronales, permiten realizar un buen uso de los datos supervisados, prescindiendo de la necesidad de considerar otros mecanismos de generalización, como son las clases de palabras, habitualmente utilizadas en esta problemática. Por otra parte, se observa que las representaciones de los términos temporales codifican conocimiento del dominio de la temporalidad, en particular información del orden y de la granularidad de las entidades. Por ejemplo, es posible reconstruir el orden de los días de la semana a partir de las representaciones.

Debido a la falta de recursos para el español al momento de iniciar esta tesis, se entrenan y evalúan representaciones vectoriales de la Wikipedia en español usando el método GloVe. Se adaptan tests de validación existentes del inglés para evaluar las representaciones, obteniéndose resultados interesantes.

Para el reconocimiento y la clasificación se consideran modelos *feedforward* y *long-short term memories* bidireccionales. Se evalúan los resultados en una partición del conjunto de entrenamiento de Tempeval 2013, obteniendo valores de 79.2% de medida F para la detección exacta y 86.1% en el caso donde se admiten corrimientos de hasta una palabra en la expresión detectada. Debido a la no disponibilidad de los datos de evaluación no es posible realizar una comparación adecuada con otros sistemas.

Finalmente, se entrenan modelos para el inglés, cuyas definiciones están orientados por las lecciones aprendidas del trabajo realizado para el español. En estos experimentos, se obtuvieron resultados de 79.1% para la detección de expresiones, resultado que está 4 puntos por debajo del estado del arte para el inglés (Lee et al., 2014).

Palabras claves: expresiones temporales, redes neuronales artificiales, semántica distribucional.

Indice

Capítulo 1

Introducción

1.1 Antecedentes

Los lenguajes naturales que utilizamos para comunicarnos son complejos sistemas que contemplan una rica diversidad de construcciones. A pesar de su complejidad, curiosamente los aprendemos con naturalidad. Las computadoras, desde su inicio, fueron creadas para ser máquinas que emulan el pensamiento, con la capacidad de resolver problemas y manipular distintos tipos de información. Sin embargo, aunque superan la capacidad humana en cuanto a cálculos exhaustivos, no es directo utilizarlas para resolver tareas que requieren de una alta asociatividad, propia de la inteligencia humana (ej. reconocer un rostro, un sonido o interpretar el lenguaje).

En lo que respecta al procesamiento del lenguaje natural se han realizado avances en múltiples áreas, entre ellas: traducción automática, reconocimiento y síntesis de voz, minería de texto, generación y análisis del lenguaje en distintos niveles. El repertorio de tareas y de enfoques para resolverlas crece conjuntamente con la cantidad de datos digitales disponibles y el poder de cómputo.

En la riqueza del lenguaje, existen diversas maneras para hacer referencia a información temporal. Las expresiones que denotan explícitamente información de temporalidad son denominadas expresiones temporales, básicamente refieren a momentos en el tiempo (ej. *hoy*) y duraciones (ej. *durante siglos*). Estas expresiones constituyen una parte del lenguaje que adopta un léxico específico cuya interpretación puede ser realizada con cantidades, puntos e intervalos en una linea de tiempo.

El tratamiento automático de expresiones temporales puede dividirse principalmente en dos tareas: detección e interpretación. La detección consiste en indicar la extensión de las

expresiones en un texto y la interpretación en especificar su significado. Aunque estas tareas son las más habituales en la comunidad, debido a la complejidad de la interpretación, se han considerado tareas que sirven de entrada para la interpretación. Por ejemplo, la clasificación de expresiones según el tipo de información que expresan o la detección de un foco temporal en caso de requerirlo para su interpretación (ej. *5 minutos después*). La detección y la interpretación de expresiones temporales son tareas de procesamiento de lenguaje natural que han sido ampliamente abordadas con una gran diversidad enfoques.

Por un lado, están los enfoques que consisten en la especificación de reglas que contemplan la composición lingüística de las distintas expresiones temporales (Grover et al., 2010; Mani y Wilson, 2000; Puscasu, 2004). Estos enfoques son conocidos como enfoques basados en reglas. Las reglas son especificadas mediante formalismos gramaticales y la calidad de las mismas impacta directamente en los resultados del sistema. Debido a la diversidad de expresiones, escribir las reglas es una tarea compleja que requiere de gran esfuerzo para abarcar un amplio espectro del lenguaje sin descuidar la calidad. Sin embargo, como las expresiones temporales complejas están compuestas de expresiones más simples y esto es posible modelarlo con los formalismos gramaticales, los enfoques basados en reglas tienen aspectos favorables para la interpretación de las expresiones.

Se han obtenido buenos resultados mediante formalismos como expresiones regulares y transductores, entre otros. Estos sistemas tienen como contra la progresiva complejidad a medida que se incrementa la cantidad de reglas y expresiones contempladas. Generalmente están altamente relacionados al idioma, por lo que puede ser complejo adaptarlos a otros idiomas. Para las construcción de las reglas es necesario tener un amplio conocimiento del lenguaje y la problemática que se pretende resolver.

Por otro lado, es posible considerar los enfoques basados en aprendizaje automático. Estos enfoques buscan aprender a resolver las expresiones temporales a partir de textos anotados con la información referente a las expresiones temporales, es decir, aprender a resolver la tarea de forma supervisada. Este enfoque evita el considerable esfuerzo de la especificación de reglas pero tiene la desventaja de requerir de una colección importante de texto anotado. La calidad del sistema depende del tamaño y calidad del conjunto de entrenamiento. En adición a lo anterior, los enfoques basados en aprendizaje automático generalmente están formulados sobre un conjunto de atributos manualmente definidos que eventualmente dependen de recursos externos (ej. analizadores lexicográficos, ontologías, etc.) que contienen conocimiento lingüístico o específico del dominio temporal. Esto lleva a tareas de integración e ingeniería de atributos agregando complejidad al enfoque.

En esta tesis se pretende estudiar la problemática referente a las expresiones temporales con modelos donde el aprendizaje es realizado puramente a partir de los datos. Es decir,

sin la especificación de reglas, ni atributos de aprendizaje automático que introduzcan conocimiento experto, ni la utilización de recursos lingüísticos con excepción de texto escrito (no supervisado) y texto con las expresiones temporales anotadas.

Los modelos considerados usan como entrada representaciones distribuidas de las palabras. Las representaciones capturan características sintácticas y semánticas de las palabras a partir de los múltiples contextos en los que ocurren en una gran colección de texto. En los últimos años, se han desarrollado métodos que, junto a las enormes cantidades de texto digital existentes hoy en día, permiten construir representaciones con resultados interesantes.

Las palabras con significados similares o relacionados tienden a tener representaciones cercanas, como consecuencia de que tienden a ocurrir en contextos parecidos. Esta es una de las características principales de las representaciones de las palabras usadas en la actualidad (otras propiedades serán introducidas más adelante). Esto permite la generalización de datos supervisados por la expansión con términos relacionados, siendo potencialmente útil para el procesamiento del lenguaje natural y en particular para tareas referentes a las expresiones temporales. Por ejemplo, si se considera una expresión como *el 19 de septiembre de 1967*, que, debido a la asociación de los términos que la componen (es decir, *septiembre* con el resto de los meses, *19* y *1967* con otras cantidades numéricas), puede ser relacionada con otras expresiones que tengan la misma forma pero distintos componentes que están paradigmáticamente relacionados con los anteriores (ej. *el 1 de enero de 1815*). Notar además, que las restricciones de dominio en las cantidades numéricas (ej. hora, día, año, etc.) pueden ser inferidas del contexto interno a la expresión además de las diferencias en las distribuciones generales de los números.

Las redes neuronales artificiales, por su naturaleza distribuida, son capaces de manipular la información embebida en las representaciones de las palabras y mediante sucesivas transformaciones de las representaciones, inferir la información referente a las expresiones temporales teniendo consideraciones sintácticas y semánticas provistas por el texto anotado, el orden lineal de las palabras y el contexto.

Este enfoque, al no incorporar información del lenguaje y la problemática explícitamente en el modelo, puede ser aplicado directamente a distintos idiomas [1] . Se trabaja principalmente para el español, lo que agrega complicaciones adicionales por la escasez de recursos pero muestra a su vez la efectividad del enfoque en un entorno limitado.

[1] Los modelos considerados incluso pueden ser aplicados a otras problemáticas que puedan formularse como detección y clasificación secuencias de palabras.

1.2 Contribución

La problemática de las expresiones temporales ha sido ampliamente abordada con una gran diversidad de enfoques. El principal aporte de la tesis es responder a la viabilidad de detectar e interpretar expresiones temporales exclusivamente de los datos y la efectividad de las representaciones distribuidas para hacerlo. Esto es importante en el sentido de la real inteligencia artificial, vincular entidades mediante representaciones internas y obtener información compuesta, sin reglas ni ingeniería de atributos.

En el transcurso de esta tesis se construyeron representaciones para las palabras y se adaptaron datos de evaluación para el español. Se realizaron evaluaciones y visualizaciones de las representaciones construidas, estas representaciones y sus resultados están disponibles para futuros trabajos de procesamiento de lenguaje y áreas afines (capítulo 5.1). En las representaciones se observaron comportamientos interesantes en cuanto a la granularidad y el orden (sección 5.2). Estas propiedades son de interés para el tratamiento de la temporalidad y son un argumento más del poder de las representaciones.

Por otro lado, se entrenan diversos modelos de aprendizaje supervisado de redes neuronales usando las representaciones construidas para el reconocimiento y clasificación de expresiones temporales, analizando el comportamiento de considerar distintas variantes en los modelos como ruido en la entrada, ventanas de contexto o contextos provistos por modelos recurrentes, entre otros. Se presentan los resultados de los modelos entrenados, mostrando el desempeño del enfoque considerado (capítulo 6).

1.3 Estructura del documento

El informe se estructura de la siguiente manera. El capítulo actual constituye la introducción y describe la contribución principal de la tesis. El capítulo 2 está destinado exclusivamente a las expresiones temporales, introduciéndolas desde un punto de vista lingüístico y comentando la diversidad de enfoques con los que han sido tratadas. El capítulo 3 introduce a las redes neuronales artificiales. El capítulo 4 está dedicado a las representaciones distribuidas de las palabras. El capítulo 5 presenta la construcción de las representaciones utilizadas, evaluándolas y mostrando propiedades referentes a las representaciones del léxico temporal, favorables para las tareas vinculadas a la interpretación de las expresiones. El capítulo 6 describe los modelos de redes neuronales considerados, utilizando las representaciones previamente construidas, y reporta los resultados obtenidos, reflejando el efecto de las distintas técnicas y consideraciones para los modelos. Finalmente, el capítulo 7 presenta las conclusiones de la tesis y posibles lineas de trabajo futuro.

Capítulo 2

Expresiones Temporales

Este capítulo introduce las expresiones temporales y las dificultades que presenta su tratamiento automático. El capítulo está dividido en dos partes; en la primera se presentan las expresiones desde un punto de vista lingüístico y en la segunda, se aborda el tratamiento de las expresiones desde un punto de vista computacional, introduciendo la diversidad de enfoques que han sido considerados para su reconocimiento e interpretación.

2.1 Expresiones Temporales

Las *expresiones temporales* son las expresiones lingüísticas que utilizamos para indicar la ubicación en el tiempo de un suceso o su duración. Considere el siguiente fragmento de texto:

- *Mañana de madrugada* se verá un cometa que pasa *una vez cada 342 años*. Hay que estar atentos, porque durará *8 minutos aproximadamente*.

En letra itálica están marcadas las expresiones temporales. La expresión *mañana de madrugada* es una expresión temporal de localización que indica un momento en la madrugada del día siguiente al actual. La siguiente expresión, *una vez cada 342 años* indica una frecuencia, es decir, un evento que se repite, y requiere de la expresión anterior para ubicar la secuencia en la linea de tiempo. Por último, la expresión *8 minutos aproximadamente* indica una duración, y es posible interpretarla como una cantidad de tiempo o el tamaño de un intervalo temporal cuyos extremos no están especificados.

Las expresiones temporales constituyen en si mismas un sublenguaje con un léxico específico que denominamos léxico temporal. Este léxico depende principalmente del sistema

de calendario que utilizamos y de convenciones culturales. Incluye palabras como los nombres de los días de la semana, los meses y los nombres de los días festivos. Algunos de los principales términos temporales son presentados en la tabla 2.1. Además de un léxico específico, las expresiones temporales, al igual que el resto del lenguaje, tienen características composicionales y permiten referir a significados similares con construcciones diferentes.

Calendario	**Días**	**Meses**	**Adverbios**	**...**
segundo	lunes	enero	hoy	
minuto	martes	febrero	mañana	
hora	miércoles	marzo	ayer	
día	jueves	abril	últimamente	
semana	viernes	mayo	ahora	
mes	sábado	junio	ya	
...	domingo	...	...	

Tabla 2.1 Tabla con algunos de los principales términos temporales.

Los términos temporales pueden agruparse en conjuntos que satisfacen determinadas propiedades. Por ejemplo, los *días de la semana*, los *meses*, los *años*, etc. tienen establecida una relación de orden según su precedencia temporal. Es posible distinguir a un conjunto particular de términos que denominamos *unidades temporales* (primera columna de la tabla 2.1) mediante las cuales se establecen: (a) restricciones numéricas para construir cantidades de tiempo o (b) restricciones con valores específicos a cada unidad para indicar localizaciones. Es interesante notar que además del orden de los valores de las unidades (ej. *marzo* $<$ *noviembre*), entre las propias unidades también es posible establecer un orden parcial según la relación de composición (ej. (*día* $<$ *mes*), pues un mes esta compuesto de días). A su vez, algunos de los conjuntos de términos son valores para las unidades temporales. Por ejemplo, los nombres de los meses, son valores de la unidad temporal mes. En síntesis, es posible establecer distintos tipos de relaciones y agrupaciones entre los terminos del léxico temporal.

Es sencillo determinar la presencia de una expresión temporal pero no siempre hay acuerdo en determinar su extensión, es decir las palabras que forman parte la expresión. Determinados factores, como la inclusión de la preposición principal o contemplar sub-expresiones en la expresión, pueden llevar a desacuerdos. Por ejemplo en la oración *en 1995 cantó en público*, considerar que la expresión está constituida por *en 1995* o simplemente por *1995*.

2.1.1 Tipo

Una distinción fundamental en las expresiones temporales es el *tipo* de información temporal que representan. Una consideración incial es si indican la *duración* o *localización* de un evento en el tiempo.

Las expresiones de localización pueden diferenciarse por consideraciones estructurales de la información que refieren. Por ejemplo, si representan un *punto* en el tiempo, un *intervalo* delimitado por sus extremos o una secuencia de puntos (Han y Kohlhase, 2003). También puede resultar conveniente distinguir las expresiones que refieren a *fechas* (día, mes, año, etc.) de las que refieren a un *momento del día* (mañana, tarde, noche).

La clasificación y nomenclatura considerada es la de TimeML (Pustejovsky et al., 2003). La misma consiste en expresiones de tipo *duración*, *fecha*, *hora* y *conjunto*. A continuación se detalla cada tipo.

Duración

Las expresiones de tipo *duración* expresan la duración de un suceso mediante una cantidad de tiempo. Suelen comenzar con la preposición *durante* seguido de la cantidad de tiempo pero existen muchas otras variantes. En la tabla 2.2 se muestran ejemplos de expresiones de este tipo junto a una porción del contexto donde ocurren.

Contexto Izq.	Expresión	Contexto Der.
estaré	durante una hora	por aquí
trabajó	por 20 días	sin descanso
me llevó	dos días completos	terminarlo
duró	diez largos años	que fueron ...
en hacerlo	una hora y media	demoró

Tabla 2.2 Expresiones de duración.

En algunos casos una expresión no determina una cantidad de tiempo con exactitud, expresiones como *poco más de tres minutos* o *algún tiempo* dan información imprecisa. En la tabla 2.3 se muestran ejemplos de expresiones con esta característica. Cabe mencionar que este fenómeno no es exclusivo de las expresiones de duración.

También se dan casos de ambigüedad, sobre todo si las expresiones son aisladas de su contexto. Puede ocurrir que no sea posible distinguir si una expresión es de duración sin hacer uso de su contexto lingüístico. En la tabla 2.4 se presenta un ejemplo donde el tipo de dos expresiones que se escriben igual, cambia dependiendo del contexto donde ocurren.

Contexto Izq.	Expresión	Contexto Der.
esperé	durante más de dos horas	hasta que ...
tardó	mucho rato	con los papeles
lo veo correr	por muchas horas	sin descanso
tardó	un santiamén	en hacerlo
fueron	varios duros años	sin noticias
demoraré	menos de 10 minutos	en llegar

Tabla 2.3 Expresiones de duración imprecisas.

Es importante notar que las expresiones de duración deben estar vinculadas a un evento que transcurre en el tiempo y en ningún caso a un evento puntual.

Contexto Izq.	Expresión
... lo hizo	en cinco minutos
... estaré de vuelta	en cinco minutos

Tabla 2.4 Ambigüedad entre duración y localización.

En algunos casos, aunque la expresión refleje una duración, puede además incluir información de la localización temporal del suceso. Por ejemplo, una expresión como *llovió durante la noche* refiere a que el suceso duró la extensión de la noche pero también que ocurrió en ella. Situaciones análogas ocurren con expresiones como *durante casi todo el mes que viene* o *durante la primera mitad del año*. Es natural considerar en esta situación a la expresión como una expresión de localización.

Fecha

Las expresiones de tipo fecha indican puntos temporales de unidad igual o superior al día. Con fechas es posible expresar determinado día, mes, bimestre, trimestre, cuatrimestre, semestre, año, década, siglo, período de la historia, etc.

Una forma de escribir fechas es mediante la especificación de sus unidades, por ejemplo *el 21 de septiembre de 1764*, pero esta es apenas la forma más directa, al considerar variantes como *el tercer jueves de mayo de 1847* o *algún mes del año siguiente* es posible apreciar la diversidad de las expresiones de este tipo. En este último ejemplo, se tienen dos particularidades interesantes, por un lado la subexpresión *del año siguiente* expresa un año que depende de una referencia temporal contenida en el contexto y por otro lado, *algún mes* agrega incertidumbre en el mes especificado. En la tabla 2.5 se presentan algunos ejemplos variados de este tipo expresiones.

Expresiones de tipo fecha
marzo de 1995
el jueves 4 de abril de 2002
abril
los primeros días de julio
la navidad pasada
a finales del verano
la edad media
el siglo XVII
el verano del 68

Tabla 2.5 Expresiones de tipo fecha.

Este tipo de expresión ofrece una rica diversidad donde un pequeño cambio en una expresión puede significar cambios importantes en su interpretación. Por ejemplo, una expresión como *el 2 de julio de 1863* puede ser resuelta con exactitud y sin necesidad de acudir a información externa. Sin embargo, al considerar la frase truncada *el 2 de julio* no es posible interpretarla como un único punto en el tiempo a menos que se disponga de la información necesaria externa a la expresión.

Hora

Las expresiones que denominamos de tipo *hora* son las utilizadas para indicar la localización temporal de un suceso en un momento del día. Hacen referencia a las unidades de granularidad menor a día. Pueden considerar la especificación de hora, minuto, segundo, etc. o momentos del día como la madrugada, mediodía, atardecer, etc.

Expresiones de tipo hora
11:55 am
21:19
las 3 de la tarde
primeras horas de mañana
2 y media de la tarde
la medianoche
cuatro menos cuarto
tres horas después

Tabla 2.6 Expresiones de tipo hora.

Aunque pueda parecer un grupo mas simple que el de las fechas, en realidad, adoptan muchas de las características de este. Por ejemplo, frases como *los primeros minutos de la*

próxima hora muestran casos de complejidad comparable a las fechas, aunque quizás sea menos frecuente encontrar estas situaciones en las expresiones de tipo *hora*, dependiendo del dominio que se trate.

Conjunto

Por último consideramos a las expresiones de tipo conjunto. Este tipo de expresiones es el utilizado para describir conjuntos de puntos en forma de frecuencias. Expresiones como *todos los lunes* o *una vez cada dos semanas* son expresiones de tipo conjunto.

2.1.2 Modo de referencia

Como se vió anteriormente, hay expresiones temporales que no contienen toda la información necesaria para su completa resolución. Expresiones como *hoy* o *al año siguiente* requieren considerar la existencia de una referencia temporal externa para su interpretación. Esta característica es independiente al tipo de la expresión.

Las expresiones que contienen toda la información necesaria para su interpretación (ej. *1998*) se denominan completamente especificadas o *absolutas*. En general estas expresiones no tienen desplazamientos y está especificada la unidad de *año* o alguna unidad de granularidad superior.

Contrariamente, las expresiones que requieren de una referencia externa se denominan *relativas* y pueden sub-clasificarse a su vez en *deícticas* o *anafóricas*, según la naturaleza de la referencia requerida. Las expresiones *deícticas* son aquellas que la referencia temporal es el momento de enunciación o de creación del documento. Expresiones como *hoy*, *mañana* y *el año que viene* son expresiones de este tipo. Por otro lado, las expresiones *anafóricas* tienen una referencia temporal a otra expresión (o evento) que fue mencionada anteriormente en el texto. Por ejemplo, en la oración "La resolución de *1989* fue un problema para *el año siguiente*.", para resolver la expresión *el año siguiente* es necesaria la expresión *1989*.

2.1.3 Precisión

La precisión refiere a si la expresión temporal denota un objeto con exactitud. La falta de precisión o incertidumbre en una expresión temporal puede tener distintos motivos. Un factor es la ocurrencia explícita de *cuantificadores indefinidos* como: *algún*, *varios*, *mucho* o *casi todos*; o modificadores como: *a comienzos de*, *a mediados de*, etc. Ejemplos de expresiones de este tipo son: *algún día de abril*, *a finales del año pasado* o *algún fin de semana a mediados de este año*.

Otra forma de introducir incertidumbre es con términos como: *rato*, *santiamén*, *período*, etc.; o expresiones como: *un tiempo* . Además es posible considerar modificadores como: *después de* o *más de* que permiten construir expresiones como: *después del lunes*, que plantea una restricción de desigualdad con la entidad temporal que la expresión refiere.

La ocurrencia de términos en plural sin determinantes es otro indicativo de imprecisión en la información denotada por la expresión. Por ejemplo, la expresión *días después* representa un desplazamiento de una cantidad no especificada de días a una referencia temporal (posiblemente con una interpretación equivalente a *algunos días después*).

2.1.4 Granularidad

Las expresiones temporales representan intervalos en una linea de tiempo. En el caso de las expresiones de duración, a diferencia de las de localización, intervalos cuya posición no está determinada. Podría establecerse una analogía entre las unidades temporales (minuto, hora, día, mes, etc.) y las unidades del sistema métrico (centímetro, decímetro, metro, etc.). Al igual que una distancia se especifica utilizando unidades métricas, los intervalos que denotan las expresiones temporales estan especificados mediante unidades de tiempo.

La granularidad de una expresión temporal corresponde a la unidad menos significativa que es especificada por la expresión. Por ejemplo, en una expresión como *1 hora y 20 minutos*, la granularidad es *minuto*; en una como *ayer*, es *día*; y en *el mes que viene*, la granularidad es *mes*.

Aunque pueda parecer un concepto simple, en algunos casos, determinar la granularidad de una expresión puede ser complejo y llevar a desacuerdos. Casos problemáticos pueden presentarse, por ejemplo, con la palabra *ahora* cuya granularidad depende del contexto. Para fijar ideas considere los siguientes ejemplos:

- Salgo para ahi *ahora*.
- En la década del 80 las computadoras no escuchaban, *ahora* las cosas han cambiado.

En el primer caso, la palabra *ahora* significa precisamente el momento actual, su granularidad podría considerarse la de un instante o momento del día. En el segundo caso, claramente la granularidad es mayor y se refiere posiblemente a la década o época actual.

2.1.5 Esquemas de Anotación

Un esquema de anotación es un mecanismo para representar determinada información contenida en uno o varios documentos de texto. En el caso de las expresiones temporales

puede consistir en anotar las expresiones y una representación de su significado. Puede contemplar además relaciones entre expresiones u otros elementos.

El esquema de anotación más utilizado para la anotación de expresiones temporales es *TimeML* (Pustejovsky et al., 2003). Consiste en etiquetas *XML* para anotar expresiones temporales, eventos y relaciones contenidos en un texto. El esquema ha sido utilizado en la competencia internacional *TempEval* (UzZaman et al., 2012). En lo que a este trabajo refiere, nos centramos en la parte del esquema destinada a la temporalidad. A continuación se presenta un fragmento de texto anotado con TimeML extraido de la competencia *TempEval*:

```
El grupo britanico se ha <EVENT eid="e1" class="I_STATE">visto</EVENT>
<EVENT eid="e2" class="I_STATE">obligado</EVENT> a <EVENT eid="e3"
class="I_ACTION"> aplazar</EVENT> los <EVENT eid="e4" class="OCCURRENCE">
conciertos</EVENT> que <EVENT eid="e5" class="I_STATE">tenia</EVENT>
<EVENT eid="e6" class="I_STATE"> programados</EVENT> para <TIMEX3 tid="t1"
type="DATE" value="2002-02-05">el proximo martes</TIMEX3> en Razzmatazz
(que se <EVENT eid="e7" class="STATE"> pospone</EVENT> <TIMEX3 tid="t2"
type="DATE" value="2002-04-25">al 25 de abril</TIMEX3>) y, <TIMEX3
tid="t3" type="DATE" value="2002-04-26">un dia despues</TIMEX3>, en
la Sala Multiusos de Zaragoza.
```

En *TimeML*, un componente central para la interpretación de las expresiones temporales es el atributo *value* de la etiqueta *TIMEX3*. En él se codifica la entidad temporal expresada en un formato que extiende al *ISO-8601*. Este atributo no indica la forma en la que la entidad temporal es obtenida sino que la especifica directamente. Esto da lugar a considerar otros esquemas de representación temporal, que permitan obtener conclusiones incluso en casos donde no sea posible determinar las entidades temporales completamente. *TCNL (por Time Calculus for Natural Language)* (Han et al., 2006; Han y Kohlhase, 2003) es un cálculo temporal para representar expresiones temporales preservando la forma en que la entidad esta siendo expresada en lenguaje natural.

En resumen, un esquema como *TimeML* está inclinado a la anotación de textos: la extensión de las expresiones, eventos y sus vinculaciones; mientras que *TCNL* se enfoca en el razonamiento temporal y en permitir una representación que preserve la esencia de la forma lingüística.

2.2 Tratamiento Automático

El tratamiento automático de las expresiones temporales es una tarea clásica del procesamiento de lenguaje natural que ha sido abordada con diversos enfoques. Dicho tratamiento presenta varias dificultades, entre las cuales se destaca:

- Tienen una rica diversidad y estructura composicional (ej. *[algún día [del verano [del año próximo]]]*).
- Puden confundirse con expresiones que no son temporales (ej. consiste *en 2000* piezas).
- Su significado puede variar dependiendo del contexto (ej. nos veremos *el martes* / nos vimos *el martes*).

Los trabajos iniciales orientados al tratamiento automático de expresiones temporales se basaban principalmente en como representar la temporalidad (Becher et al., 1998) y en la construcción de mecanismos de representaciones que faciliten la inferencia automática en textos (Mani y Wilson, 2000). A la fecha, se han desarrollado diversos sistemas que usan diferentes formalismos y técnicas que pueden clasificarse, según la naturaleza de sus decisiones, basados en *reglas* o en *aprendizaje automático*. Los sistemas basados en reglas contienen manualmente codificado el conocimiento experto de las expresiones, mientras que los basados en aprendizaje buscan aprenderlo de ejemplos con métodos de aprendizaje supervisado. Los sistemas que combinan técnicas de aprendizaje con reglas y conocimiento experto, se denominan sistemas híbridos, por ejemplo, los *parsers* semánticos probabilísticos donde una parte del léxico y las reglas son construidos manualmente y el criterio de selección del análisis está basado en las ocurrencias de las expresiones en textos.

2.2.1 Enfoques basados en Reglas

Los sistemas basados en reglas se construyen con restricciones y patrones que especifican la forma de las expresiones temporales reconocidas, valiéndose además de la composicionalidad de las reglas para interpretar a las expresiones. Las gramáticas categoriales, expresiones regulares y transductores son algunos de los mecanismos utilizados por este tipo de enfoque.

Este enfoque tiene la ventaja de expresar directamente la forma de las expresiones, teniéndose control sobre los resultados y la cobertura de expresiones reconocidas. Los errores pueden ser corregidos minuciosamente, por lo cual es posible construir sistemas precisos con buenos resultados. Sin embargo, esto puede tener el costo de definir un amplio repertorio de reglas y léxico específico para la tarea.

A pesar de la complejidad de construir un sistema de reglas, con determinadas consideraciones es posible abarcar un espectro amplio de expresiones temporales con relativamente pocas reglas. A modo de ejemplo considere el formalismo de las gramáticas libres de contexto, donde los símbolos terminales son las palabras del texto. Es posible abstraer términos del dominio mediante reglas como:

$$\underline{mes} \rightarrow enero, febrero, ..., diciembre$$

$$\underline{día} \rightarrow lunes, martes, ..., domingo$$

obteniendo clases de términos bajo ciertas variables (términos subrayados) que son utilizadas en construcciones compuestas. Por ejemplo,

$$timex\_fecha\_a \rightarrow el\ \underline{día}\ \#num\ de\ \underline{mes}\ de\ \#num$$

abarca todas las expresiones con dicha forma (ej. *el martes 9 de abril de 2047*) . A su vez, considerando operadores de opcionalidad y selección en el lado derecho de las reglas, aumenta el poder de abstracción y disminuye la cantidad de reglas necesarias para construir el sistema.

En situaciones donde es necesario acudir al contexto de largo alcance o consideraciones semánticas para desambiguar una expresión puede ser problemático el uso de reglas, al igual que proveer un índice de confianza de los resultados obtenidos. Además, estos sistemas al especificar la composición de las expresiones son en general muy dependientes del lenguaje tratado. Esto implica que portar un sistema basado en reglas de un lenguaje a otro puede tener diversas dificultades y requerir la reescritura de las reglas. Cabe mencionar que la mayoría de los sistemas desarrollados son para el inglés.

En cuanto a sistemas existentes basados en reglas, Mani y Wilson (2000) presentan un sistema que resuelve el reconocimiento y la interpretación mediante reglas manualmente definidas y otras aprendidas con aprendizaje automático. Puscasu (2004) trata el problema mediante una secuencia de procesamientos del texto de entrada, donde cada etapa agrega acumulativamente información de las expresiones mediante reglas y heurísticas. Grover et al. (2010) resuelven el reconocimiento y la interpretación mediante un formalismo de reglas sobre la información morfosintáctica de la entrada. Filannino (2012) construye un sistema utilizando como base el sistema *TRIOS* (UzZaman y Allen, 2010), agregando etapas de pre y pos-procesamiento para mejorar sus resultados.

El sistema HeidelTime (Strötgen y Gertz, 2010) utiliza expresiones regulares y recursos del léxico temporal para el reconocimiento y la interpretación de las expresiones, obteniendo los mejores resultados en *TempEval-2* (86% de medida F). Estos resultados fueron superados

posteriormente por el sistema SUTime (Chang y Manning, 2012). En la *TempEval-3*, el sistema que obtuvo mejores resultados fue una nueva versión de *HeidelTime* (Strötgen y Gertz, 2013). Bethard (2013b) supera este resultado con un sistema basado en una gramática libre de contexto manualmente desarrollada.

2.2.2 Enfoques basados en Aprendizaje

Los métodos clásicos de aprendizaje automático consisten principalmente en clasificadores basados en conjuntos de atributos definidos convenientemente. A partir de ejemplos anotados, se establecen mecanismos para determinar la información deseada en entradas arbitrarias. Este tipo de métodos es adecuado para la identificación de expresiones temporales pero no es directa su aplicación para la interpretación.

Estos sistemas generalmente se basan en atributos como la categoría de palabras en una ventana de contexto, la pertenencia a clases manualmente especificadas de palabras o restricciones sobre un análisis de dependencias. La diversidad de atributos posibles es ilimitada. Los resultados dependen significativamente de los atributos utilizados. También es importante notar que atributos relevantes pueden consumir un tiempo de cómputo considerable haciendo que su uso sea cuestionable.

Es posible mencionar muchos sistemas existentes basados en aprendizaje automático. Adafre y de Rijke (2005) efectúan la detección de expresiones temporales con *Conditional Random Fields (CRF)*, al igual que Ahn et al. (2005). Luego, Ahn et al. (2007) utilizan *Support Vector Machine (SVM)* como clasificador y simplifican las reglas para la interpretación mediante una cascada de clasificadores.

Enfoques semi-supervisados pueden incluir técnicas de *bootstrapping* para mejorar el reconocimiento (Poveda et al., 2009). Kolomiyets y Moens (2010) realizan la expansión de casos positivos con *WordNet*. El sistema *ManTIME* (Filannino et al., 2013) efectúa la detección con clasificadores *CRF*, considerando atributos derivados de *WordNet* (sin obtener mejoras significativas).

El sistema *ClearTK-TimeML* (Bethard, 2013a) entrena múltiples clasificadores supervisados para identificar y clasificar expresiones temporales, eventos y relaciones. El sistema hace competir distintos métodos (*CRF*, *SVM* y regresión logística) aplicando determinada selección de hiper-parámetros.

2.2.3 Enfoques Híbridos

Técnicas de aprendizaje automático y formalismos de reglas han dado buenos resultados en el reconocimiento. En cuanto a la interpretación, las reglas se utilizan con cierta naturaliad

por sus propiedades composicionales. Algunos enfoques combinan las ventajas de los formalismos de reglas con ejemplos anotados para la interpretación de expresiones. Angeli et al. (2012) infieren una gramática libre de contexto probabilística sobre las expresiones. Este sistema es aplicable fácilmente en distintos idiomas (Angeli y Uszkoreit, 2013).

Lee et al. (2014) introducen un sistema que utiliza gramáticas categoriales combinatorias para la detección e interpretación de las expresiones temporales. El trabajo considera un léxico manualmente construido de 287 entradas léxicas y otras generadas automáticamente (como números y formatos de fechas), obteniendo resultados de 83.1% de medida F para la detección, 85.4 para la clasificación y 82.4 para la interpretación en los datos de evaluación de *TempEval-3*. Este resultado constituye el estado del arte hasta el momento.

Los enfoques presentados introducen conocimiento específico del problema (manualmente) mediante la definición de reglas, atributos de aprendizaje automático y clases de palabras, entre otros. Esto vuelve al enfoque altamente dependiente del problema tratado y el lenguaje en el que se trabaja. En lo que sigue se introducen las redes neuronales artificiales y las representaciones distribuidas de las palabras. Estos conceptos permiten resolver tareas vinculadas a las expresiones temporales sin ninguna información específica del problema a excepción del texto anotado, utilizado para entrenar los modelos, y el texto (no anotado) con el que se construyen las representaciones de las palabras.

Capítulo 3

Redes Neuronales Artificiales

3.1 Conceptos Generales

Las Redes Neuronales Artificiales son modelos matemáticos relacionados con el funcionamiento del cerebro biológico (Rosenblatt, 1963). Los modelos se basan en la existencia de unidades de procesamiento capaces de recibir múltiples entradas y emitir una salida. Las unidades están interconectadas a través de sus entradas y salidas, con parámetros a ajustar en las conexiones. En analogía con el sistema nervioso, las unidades son neuronas, las entradas corresponden a las dendritas, la salida a la información emitida por al axón y cada parámetro de conexión corresponde a la intensidad de dicha sinapsis.

Los modelos neuronales son capaces de ajustarse para transformar determinada entrada, dada por una representación vectorial, en una salida con cierta finalidad. La entrada es recibida por unidades específicas, denominadas unidades de entrada, que se activan y propagan la información a otras neuronas a través de las conexiones. La información se propaga activando unidades hasta llegar a las unidades de salida, que en conjunto forman la respuesta de la red. Este proceso es relacionado con la actividad sináptica del cerebro.

Han surgido diversas variantes y aplicaciones de modelos neuronales y aunque es cuestionable su relación con el cerebro biológico, han mostrado buen desempeño y flexibilidad en tareas de aprendizaje automático modelando clasificadores y regresiones. Además, son modelos interesantes por su relación natural con las representaciones distribuidas.

3.2 Modelo Feed-forward

El modelo *feed forward* es la base de los modelos considerados en esta tesis. Su característica principal es la inexistencia de ciclos entre las neuronas. El modelo *feed forward* más simple es el *perceptrón* (Rosenblatt, 1958), que corresponde a una unidad de procesamiento que realiza la combinación lineal de la entrada con los parametros a ser ajustados y retorna una salida binaria indicando si el resultado es positivo.

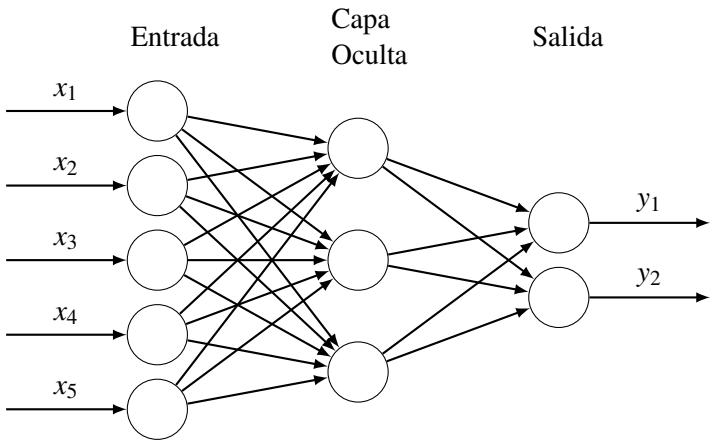

Fig. 3.1 Ejemplo de red neuronal *feed forward.*

En modelos como el *perceptrón multicapa* (Rumelhart et al., 1986) las unidades se organizan en capas. Las neuronas de una capa reciben como entrada la salida de las unidades de la capa anterior y no hay conexiones dentro de la misma capa. La figura 3.1 presenta el diagrama de una red *feedforward.*

Las unidades neuronales invocan una función no lineal sobre una combinación lineal de las entradas denominada *función de activación* (fig. 3.2). Haciendo referencia al impulso sináptico se usa una función con forma de escalón. Se han considerado muchas alternativas de función de activación, dos habituales son la función sigmoide y la tangente hiperbólica. Se consideran funciones que sean diferenciables respecto a los parámetros para poder ajustarlos usando *descenso por gradiente*. Para ajustar los parámetros se define una *función objetivo* que mapea el resultado de la red a un valor que indica su rendimiento para resolver la tarea. El ajuste sistemático de los parámetros para que el modelo resuelva la tarea adecuadamente constituye el *entrenamiento* de la red.

$$h_j^{(i)} = \sigma(\sum_k w_{jk}^{(i)} h_k^{(i-1)} + b_j^{(i)}) \tag{3.1}$$

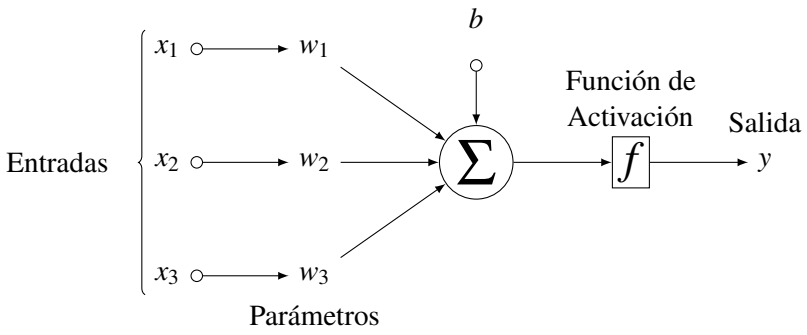

Fig. 3.2 Modelo de una neurona artificial.

La ecuación 3.1 corresponde a una capa oculta con función de activación σ, el supra índice refiere al índice de la capa y $W^i = ((w_{jk}^{(i)}))$ es la matriz de parámetros y $b^{(i)}$ el vector de desplazamientos de la capa. Las capas ocultas en este trabajo utilizan como *función de activación*, salvo que se indique lo contrario, a la función lineal rectificada ($\sigma(z) = max(0,z)$).

3.2.1 Backpropagation

Un avance fundamental en el éxito de las redes neuronales multicapa fue aprender a entrenarlas mediante la propagación del error a través de las capas en el sentido inverso a la activación de la red, desde la capa de salida hacia la capa de entrada. Se construyen los gradientes por capa mediante la aplicación de la regla de la cadena sobre las derivadas parciales, pues la transferencia de información entre capas es una composición de funciones. A esta técnica se le denomina *backpropagation* (Rumelhart et al., 1986).

La técnica *backpropagation* es la base de técnicas posteriores. Una técnica ampliamente usada en conjunto con *backpropagation* y descenso por gradiente es la consideración de *momento* en el entrenamiento. Esto es, considerar para ajustar los parámetros, además del gradiente del error actual, al gradiente del error del paso de entrenamiento anterior, lo cual permite escapar de pequeños mínimos locales (Plaut et al., 1986).

3.2.2 Softmax

Es posible definir modelos neuronales para clasificar la entrada respecto a un conjunto predefinido de n clases. En estos casos es habitual considerar la función *softmax* (Bridle, 1990) para las unidades de la capa de salida. A cada clase posible le corresponde una neurona de la capa de salida, encargándose de retornar la probabilidad correspondiente a esa clase. La salida del modelo es un vector $s =< s_1, .., s_n >$ donde $s_i \in [0,1]$ y $\sum s_i = 1$. Cada componente

es un estimador de la probabilidad de la clase asociada a esa neurona. La función de *softmax* se define como

$$s_i = \frac{e^{w_i \dot{h} + b_i}}{\sum_{j=1}^{n} e^{w_j \dot{h} + b_j}}, \tag{3.2}$$

donde w_i es el vector de parámetros de la i-ésima unidad de salida y b_i su correspondiente término independiente, ambos ajustados durante el entrenamiento; h es la entrada de la unidad, es decir, la salida de la capa oculta más próxima a la capa de salida o la entrada del modelo, en caso de que el modelo no tenga capas ocultas. La cantidad de clases de la clasificación define la cantidad de filas de W y la dimensión de b. La cantidad de columnas de W está dada por el tamaño de la entrada de la capa de salida. Notar que la función de *softmax* es diferenciable, esto permite el entrenamiento de la red con *backpropagation*.

3.3 Modelo Recurrente

La consideración de ciclos en las conexiones de los modelos permite en una activación tener en cuenta activaciones anteriores. Esto otorga al modelo una memoria dinámica en su uso secuencial (Elman, 1990). Cuando un modelo tiene ciclos entre las conexiones de sus unidades se le denomina *recurrente*. Se considera como base las redes recurrentes con ciclos de una capa oculta en si misma.

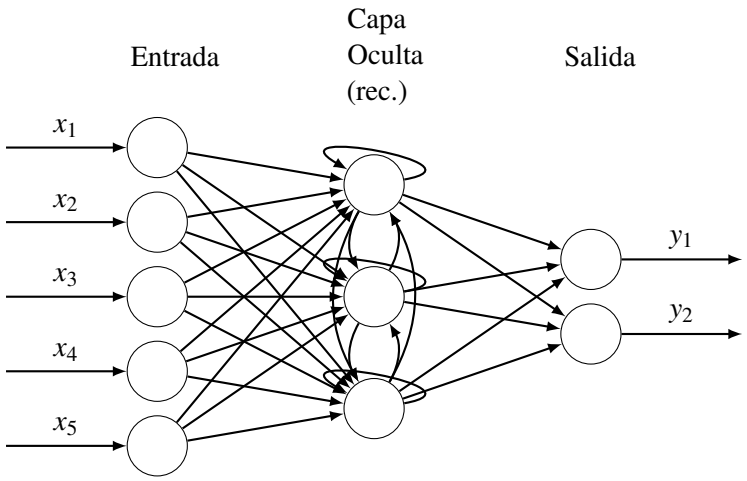

Fig. 3.3 Ejemplo de red neuronal *recurrente*.

La activación de una red recurrente es similar a la de un modelo *feed forward* incluyendo, en la capa oculta, la salida de la activación anterior. Podemos considerar que se tiene una red *feed forward* con una capa oculta distinta en cada activación, a esto se le llama despliegue (*unfolding*) del modelo y se ilustra en la figura 3.4.

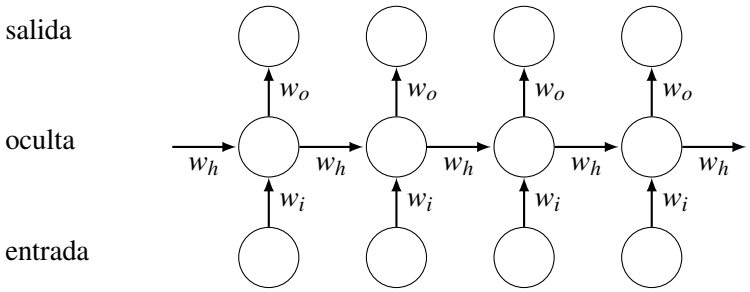

Fig. 3.4 Despliegue de una red neuronal recurrente.

La ecuación de la j-ésima unidad de la i-ésima capa oculta recurrente simple es

$$h_j^{(i)} = \sigma(\sum_l w_{jl}^{(i)} \tilde{h}_l^{(i)} + \sum_k w_{jk}^{(i)} h_k^{(i-1)} + b_j^{(i)}), \tag{3.3}$$

donde $\tilde{h}_l^{(i)}$ es la salida de la l-ésima unidad de la misma capa en la activación anterior, $h_k^{(i-1)}$ la salida de la k-ésima unidad de la capa anterior (en la activación actual) y $W^{(i)}$ y $b^{(i)}$ los parámetros a ser ajustados del modelo.

El entrenamiento de los modelos recurrentes es más complejo que el de los *feedforward* debido a la necesidad de desarrollar las activaciones a través del tiempo para calcular el gradiente. Dos técnicas desarrolladas para calcular el gradiente de las redes recurrentes son *real time recurrent learning (RTRL)* (Robinson y Fallside, 1987) y *backpropagation through time (BPTT)* (Williams y Zipser, 1995). El entrenamiento de los modelos recurrentes considerados en esta tesis se realiza con una técnica basada en la segunda de ellas.

3.3.1 Desvanecimiento del gradiente

Los modelos recurrentes son difíciles de entrenar adecuadamente. Dos problemas que presenta su entrenamiento son la explosión y el desvanecimiento del gradiente [1] (Bengio et al., 1994). El problema de explosión del gradiente se puede resolver atenuando el tamaño del gradiente en ciertas condiciones (Pascanu et al., 2012)[2], pero el problema de desvanecimiento requiere un tratamiento distinto.

[1] En inglés los términos usados son *gradient exploding* y *gradient vanishing*.
[2] La técnica se denomina *gradient clipping*.

El problema de desvanecimiento del gradiente no refiere a que el gradiente disminuye progresivamente de tamaño en su totalidad como el nombre lo sugiere, sino que ocurre durante el entrenamiento que algunas componentes se desvanecen. Esto tiene el efecto de que la red considera satisfactoriamente el contexto muy cercano sin ser capaz de tener consideraciones de contextos de mayor alcance.

Una exitosa propuesta para combatir el problema de desvanecimiento del gradiente, mediante una reparametrización de las unidades de la capa oculta, es el modelo *Long Short-Term Memory* (LSTM) (Hochreiter y Schmidhuber, 1997). Esta alternativa, aunque evita el desvanecimiento del gradiente ha sido cuestionada por la aparente innecesidad de algunos de sus componentes omitiéndolos sin degradar los resultados (Greff et al., 2015).

Una alternativa similar a las LSTM, que siendo mas simple arroja resultados parecidos, son las *Gated Recurrent Unit* (GRU) (Cho et al., 2014). Aunque la mayor simpleza de las GRU frente a las LSTM es difícil de justificar, las GRU dan resultados comparables e incluso se han reportado resultados superiores a las *LSTM* en algunos casos para un conjunto de tareas (Chung et al., 2014).

Se cuestiona la existencia de otras reparametrizaciones para las unidades de las capas ocultas recurrentes que mejoren los resultados, o al menos, que siendo más eficientes brinden resultados similares. Jozefowicz et al. (2015) buscan variantes mediante mutaciones y reportan casos de variantes que dan mejores resultados que ambas propuestas en algunas tareas.

3.4 Modelo Bidireccional

Una contra de los modelos recurrentes en una tarea de etiquetado de secuencias es que solo contemplan el contexto pasado. Esto puede degradar gravemente los resultados en tareas donde ambos contextos contienen información relevante. Una alternativa para considerar además el contexto futuro son los modelos *bidireccionales* (Schuster y Paliwal, 1997).

Un modelo recurrente bidireccional presenta la entrada a dos capas ocultas recurrentes, una en sentido directo y a la otra en el sentido inverso. Las salidas de ambas capas ocultas están conectadas a la siguiente capa. Esto brinda al modelo información del contexto pasado y futuro en la clasificación. En la figura 3.5 se presenta el despliegue en el tiempo de una red recurrente bidireccional. Estos modelos no son apropiados para tareas donde no se cuenta con información del futuro al momento de activar la red, como son las aplicaciones de tiempo real [3] .

[3] Un ejemplo puede ser un automovil que conduce con autonomía hasta un determinado destino, a partir de la información brindada por camaras de video posicionadas en el vehículo.

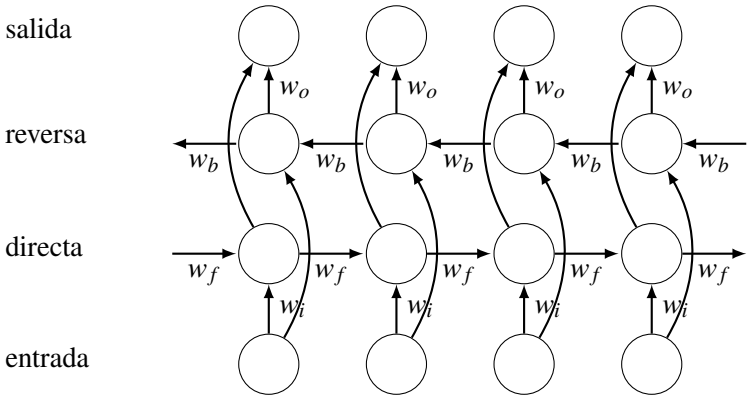

Fig. 3.5 Despliegue de una red neuronal recurrente bidireccional.

Los modelos bidireccionales pueden considerarse con unidades recurrentes como los *LSTM*. Modelos de este tipo han sido considerados satisfactoriamente en tareas de procesamiento del lenguaje. Graves y Schmidhuber (2005) utilizan *LSTMs* bidireccionales para el reconocimiento del habla. Liwicki et al. (2007) utilizan estos mismos modelos para el reconocimiento de texto en manuscrito. Un trabajo con relativa similitud al presentado en esta tesis es el de Irsoy y Cardie (2014), donde se utilizan modelos bidireccionales con representaciones vectoriales de las palabras. En este trabajo se resuelve la tarea de detección de opiniones etiquetando las palabras mediante el esquema *BILOU*, y se realiza una comparación con un modelo basado en *Conditional Random Fields* (CRF), reportándose mejores resultados para los modelos bidireccionales.

Capítulo 4

Representaciones Distribuidas de las Palabras

En este capítulo se presentan los fundamentos y modelos para la construcción de representaciones vectoriales de las palabras a partir de la distribución de los contextos en los que ocurren. Se comentan brevemente algunos modelos basados en conteo y en predicción para dar un panorama de ambos enfoques a parte de presentar *GloVe* (Pennington et al., 2014), el modelo utilizado para construir las representaciones en español. Se concluye comentando resultados que permiten entender mejor los modelos para construir representaciones.

4.1 Marco Teórico

La noción de representación distribuida puede explicarse por oposición al caso localista extremo. En un esquema de unidades[1] que pueden estar activas o no, en la teoría localista extrema cada concepto es representado mediante la activación de una unidad específica y las restantes unidades apagadas. Esta representación es denominada *one-hot encoding*. Los conceptos no tienen estructura interna y la información está basada en las relaciones entre conceptos.

En cambio, en las representaciones distribuidas, los conceptos son representados por patrones de activación en las unidades. Es decir, cada concepto se representa por un conjunto de neuronas activas y las restantes apagadas. Las representaciones distribuidas pueden permitir la generalización automática (Hinton, 1986). Además, admiten representaciones más compactas que el caso localista en cuanto a la cantidad de unidades necesarias para representar la misma cantidad de conceptos.

[1] Las unidades pueden ser neuronas de una red neuronal artificial.

Concepto	Repr.
mesa	100000...0
silla	010000...0
cama	001000...0
perro	000100...0
gato	000010...0
...	

Tabla 4.1 Representación localista de conceptos.

Concepto	Repr.
mesa	010
silla	011
cama	001
perro	101
gato	110

Tabla 4.2 Representación distribuida de conceptos.

Las representaciones distribuidas permiten la recuperación de conceptos mediante patrones parcialmente especificados y dan lugar a la agrupación de conceptos mediante patrones en subconjuntos de unidades e incluso a la jerarquización. También es posible considerar inversión de patrones para representar nociones de oposición y representaciones de intensidad.

La consideración de valores continuos en las unidades da la expresividad de los espacios vectoriales al conjunto de representaciones. Los conceptos pueden interpretarse como puntos en un espacio vectorial y la noción de similitud entre conceptos se corresponde con distancias entre vectores.

En este punto emergen una serie de preguntas: Cuántas unidades se necesitan para tener representaciones adecuadas? A qué corresponde cada unidad? Cómo se determinan los valores de las unidades para cada representación? Qué significa que un conjunto de representaciones sea adecuado?

Sahlgren (2006) comenta en su tesis que las representaciones distribuidas de conceptos tienen sus origenes en la psicología, precisamente, con el enfoque semántico diferencial (Osgood et al., 1957; Osgood, 1952). El enfoque consiste en representar los conceptos mediante la asignación de valores (en una escala de siete puntos) a un conjunto de pares de adjetivos contrastivos como pequeño-grande, bajo-alto, rápido-lento, etc. Enfoques como este permite tener nociones de distancia entre los conceptos pero aún es incierto la cantidad de unidades necesarias, como escoger los valores de cada unidad, e incluso, es deseable contar con un mecanismo para construir las representaciones automáticamente.

Un camino para la construcción automática de representaciones distribuidas de las palabras reside en disponer de una distribución de información relativa a cada palabra, que las vincule entre ellas. En particular, es posible representar el significado de las palabras mediante su uso. La *hipótesis distribucionalista* sostiene que las palabras con tendencia a ocurrir en contextos similares tienen significados similares (Harris, 1954). Generalizando esta hipótesis, se considera el significado de las palabras a través de la distribución de sus contextos, mediante la ampliamente mencionada cita: "*You shall know a word by the company it keeps*" (Firth, 1957).

La distribución de los contextos de una palabra es el conjunto de los contextos donde esa palabra ocurre. Esto puede ser llevado a la práctica considerando las palabras próximas a las ocurrencias de cada palabra en una colección de texto suficientemente grande. Bajo esta suposición diversos métodos para construir representaciones distribuidas de las palabras han sido desarrollados. Por un lado están los métodos basados en conteos de frecuencias de coocurrencias de palabras. Por otro lado están los métodos basados en predicción, que infieren las representaciones mediante el ajuste progresivo de los valores de las unidades mediante una función a optimizar. Este último es el caso de los modelos basados en redes neuronales artificiales.

La consideración de una ventana de contexto tiene la ventaja de la simplicidad y es ampliamente utilizado, asimismo, otras definiciones de contexto han sido consideradas. Una alternativa que es interesante mencionar se encuentra en la consideración de contextos brindados por un análisis de dependencias. Levy y Goldberg (2014a) muestran que las representaciones construidas con los contextos tomados de las dependencias entre las palabras dan interesantes resultados, distintos a los obtenidos con los contextos provistos por ventanas de palabras.

4.2 Modelos basados en conteo

La información disponible y habitualmente usada para construir representaciones para las palabras son las colecciones grandes de texto. El texto brinda distribuciones de contextos para las palabras y las representaciones se construyen mediante la consideración conjunta de las mismas. Los modelos basados en conteo se basan directamente en información de las frecuencias de las palabras y sus contextos.

Como modelo introductorio a este enfoque, es posible considerar una mátriz $M = ((m_{i,j}))$ de tamaño $|V| \times |C|$, donde V es el conjunto de palabras del texto (vocabulario), $|C|$ el conjunto de contextos donde puede ocurrir cada palabra y $m_{i,j}$ corresponde a la cantidad de ocurrencias de la palabra i en el contexto j. Aún queda por definir cuáles son los contextos

posibles. Una alternativa es tomar a $|C|$ como el conjunto de n-gramas del texto. En este caso, el resultado y dimensión de la representación depende del n utilizado. Un valor para n pequeño va a llevar a representaciones más compactas mientras que un valor mayor brinda contextos más informativos.

En el enfoque planteado se puede apreciar que muchas de las decisiones tomadas son arbitrarias y es posible considerar una infinidad de variantes. La medida entre palabras y contextos puede ser más informativa que la frecuencia, otras formas para los contextos, tratamientos al texto previos a realizar los conteos, transformaciones a la matriz, son algunas de ellas. Distintas consideraciones dan lugar a distintos resultados pudiendo impactar drásticamente en la calidad de las representaciones obtenidas. Al final de este capítulo se volverá sobre este punto (seccion 4.4.2).

En lo que sigue de esta sección se presentan algunos de los modelos de conteo más influyentes.

4.2.1 Análisis Semántico Latente

Análisis Semántico Latente (LSA, por su nombre en inglés *Latent Semantic Analysis*) fue de los primeros métodos reportados en mostrar resultados interesantes en cuanto a capturar el significado de las palabras basándose en el conteo de las ocurrencias de las palabras y reducciones de dimensión con técnicas de factorización de matrices (Deerwester et al., 1990).

Versiones iniciales de *LSA* consideran una matriz término-documento capturando las ocurrencias de los términos en los documentos. El enfoque se relacionaba con el problema de descubrir los tópicos de un conjunto de documentos. Las filas de la matriz se corresponden con los términos y las columnas con los documentos de la colección. Sean $T = t_1, ..., t_n$ el conjunto de términos y $D = d_1, ..., d_n$ el de documentos. La matriz $X = ((x_{ij}))$ de dimensión $n \times m$ corresponde a la relación término-documento. En planteos iniciales, x_{ij} se definió como la cantidad de ocurrencias del término t_i en el documento d_j pero se han considerado diversas alternativas con mejores resultados.

La matriz X es factorizada usando *SVD (Singular Value Decomposition)*, esto permite obtener representaciones más compactas. La matriz X se descompone como

$$X = T_0 S_0 D_0 \tag{4.1}$$

donde T_0 y D_0 son matrices ortogonales y S_0 es una matriz diagonal que contiene los valores singulares en orden decreciente. De esta descomposición se obtiene una representación vectorial para los términos y otra para los documentos. La descomposición brinda representaciones más compactas. La dimensión de las representaciones está dada por la dimensión de

la matriz de valores singulares. En principio el tamaño de S_0 es $d \times d$ donde d es el rango de X, pero es posible reducir la dimensión de los espacios de representación truncando la descomposición a los k mayores valores singulares.

4.2.2 Hiperespacio Análogo al Lenguaje

Lund y Burgess (1996) presentan, HAL (por *Hyperspace Analogue to Language*) , un método basado en las coocurrencias palabra-palabra para construir un espacio semántico. El procedimiento consiste en desplazar una ventana constituida por un conjunto de palabras consecutivas en el texto. Las palabras dentro de la ventana se consideran coocurrentes con una intensidad inversamente proporcional a la distancia entre ellas. Se acumulan los valores de intensidad de coocurrencia de las palabras en una matriz indexada en filas y columna por las palabras. El orden de los pares influye y por lo tanto la matriz no es necesariamente simétrica.

La matriz generada es de tamaño $n \times n$, asumiendo n palabras a representar, conteniendo una fila y una columna por cada palabra. El vector fila de una palabra revela información de las coocurrencias con palabras posteriores, mientras que el vector columna con palabras anteriores. Ambos vectores son concatenados generando un vector de dimensión $2n$. Este espacio puede resultar de dimensión extremadamente grande y se sugiere reducirla mediante *PCA (Principal Component Analysis)*.

Es posible considerar distintos valores para el tamaño de la ventana. Los autores reportan los resultados con distintos tamaños obteniendo los mejores con una ventana de 8 palabras. Es interesante comentar que la similitud considerada por los autores fue basada en las métricas $d_r(x,y) = \sqrt[r]{\sum |x_i - y_i|^r}$ (Minkowski), realizando experimentos con valores de r de 1, 1.5 y 2. Con $r = 1$ se obtuvieron los mejores resultados.[2]

4.2.3 Matriz de PPMI

Métodos como LSA y HAL mostraron que las representaciones basadas en valores estadísticos de las ocurrencias de las palabras y técnicas de reducción de la dimensión dan resultados prometedores. Es posible mejorar las representaciones considerando medidas más informativas que la cantidad de coocurrencias.

La medida PMI (por *Pointwise Mututal Information*) mide la relación entre la probabilidad conjunta entre dos sucesos y su probabilidad asumiendo independencia. Se define como:

$$pmi(w_1, w_2) = log \frac{p(w_1, w_2)}{p(w_1)p(w_2)} = log \frac{p(w_1|w_2)}{p(w_1)}, \tag{4.2}$$

[2]En la actualidad la distancia coseno es habitualmente usada por sus ventajas computacionales.

donde w_1 y w_2 son dos palabras y la probabilidad de una palabra habitualmente se calcula mediante las repeticiones de la palabras en una colección grande de texto. La medida PMI toma valores positivos en caso de correlación alta, 0 si hay independencia y negativos si la ocurrencia de uno tiende a limitar la ocurrencia del otro.

Una variante útil de la medida *PMI* es considerar cero a los valores negativos ya que no corresponden a pares que presentan correlación. Esta medida es denominada *Positive PMI (PPMI)*. Las representaciones construidas a partir de la medida PPMI en conjunto con la distancia coseno han mostrado mejores resultados que otras configuraciones, incluso que la original PMI (Bullinaria y Levy, 2007).

4.3 Modelos basados en predicción

Los modelos basados en predicción utilizan el contexto para ajustar una función de donde se obtiene una representación distribuida para las palabras. Las redes neuronales artificiales han sido ampliamente usadas en este sentido.

4.3.1 Modelos de lenguaje

Un modelo de lenguaje, en el procesamiento de lenguaje natural, permite predecir la siguiente palabra de un oración dadas las anteriores. Esto es equivalente a calcular la probabilidad de una oración usando la regla de la cadena

$$p(w_1..w_n) = \sum_i p(w_i|w_1..w_{i-1}).$$

Los modelos de lenguaje son ampliamente usados en tareas como traducción automática y reconocimiento de voz, donde es útil tener en cuenta la adecuacidad de la respuesta generada. En traducción automática por ejemplo, un modelo del lenguaje destino, permite descartar traducciones candidatas por hacer un uso adecuado del lenguaje.

Es posible construir modelos de lenguaje con redes neuronales. Estos modelos superan significativamente a los modelos clásicos basados en n-gramas. Además permiten construir representaciones distribuidas.

El modelo neuronal más simple posiblemente sea la red con una capa oculta que recibe la palabra anterior representada en *one-hot* y predice la palabra actual. La red captura en la capa oculta representaciones distribuidas para las palabras. De hecho la capa oculta opera como una capa de proyección de la representación de la palabra cuyos parámetros son aprendidos durante el entrenamiento.

Uno de los primeros modelos de lenguajes neuronales presentados fue el de Bengio et al. (2003). El modelo tiene una capa de proyección para las palabras, una capa oculta no lineal y la capa de salida. La entrada consiste en N palabras (en representación *one-hot*) que son pasadas a la capa de proyección (compartida) cuyos parámetros son las representaciones. La concatenación de las representaciones es la entrada de la capa oculta, cuya salida es utilizada para obtener la distribución de probabilidad de ser la próxima palabra según el contexto dado.

4.3.2 Aprendizaje Multitarea

Las redes neuronales, además de ser capaces de construir representaciones distribuidas, permiten considerar múltiples capas de salidas y entrenar cada salida según objetivos distintos. Algunas salidas pueden estar basadas en modelos de lenguajes y otras en tareas de lenguaje.

Esta flexibilidad de las redes neuronales fue adoptada por Collobert y Weston (2008) y Collobert et al. (2011), donde se presentan modelos de redes neuronales entrenados para resolver un conjunto de tareas del lenguaje. La arquitectura incluye un modelo de lenguaje y las tareas supervisadas: etiquetado léxico, reconocimiento de entidades con nombre, *chuncking* y etiquetado de roles semánticos. El trabajo muestra que la consideración del modelo de lenguaje lleva a que se obtengan mejores resultados en las otras tareas, que llegan a ser comparables con sus respectivos estados del arte.

4.3.3 Skip-Gram y CBOW

Mikolov et al. (2013a) presentan dos métodos para construir representaciones distribuidas de las palabras: *skip-gram* y *cbow* (por *Continous Bag of Words*). Ambos son modelos neuronales que usan el contexto local de las palabras en una gran colección de texto.

El modelo de *cbow* es una red neuronal con una capa compartida de proyección en la entrada. No tiene capa oculta y la capa de salida devuelve una distribución de probabilidad sobre las palabras del vocabulario. La red predice la palabra central de una ventana de palabras. Por otro lado, *skip-gram*, a partir de una palabra predice su contexto. Dada una secuencia de palabras $w_1..w_n$ la función objetivo a maximizar es

$$\sum_{i=1}^{n} \sum_{i-c \leq j \leq i+c, j \neq i} p(w_j|w_i). \tag{4.3}$$

Por cuestiones de eficiencia no es adecuado definir la distribución de probabilidad $p(w_j|w_i)$ con la función de *softmax*, pues el costo de calcular el gradiente depende del tamaño del vocabulario que es del orden de los cientos de miles de palabras. Para tratar este

problema, Mikolov et al. (2013b) consideran dos alternativas: *hierchical softmax* y *negative sampling*.

En el *hierchical softmax* se aproxima la función con una estructura de árbol con las palabras en las hojas reduciendo el costo a logarítmico. Mientras que *negative sampling* se basa en la idea de construir representaciones capaces de diferenciar los ejemplos de entrenamiento de casos corruptos generados artificialmente. La función objetivo considera k casos corruptos creados a partir del ejemplo para contrastarlo.

La función objetivo de *skip-gram* con *negative sampling* consiste en considerar para $p(w_j|w_i)$ en la ecuación 4.3 a

$$\log \sigma(\tilde{v}_{w_j}^T v_{w_i}) + \sum_{w_i \in S_k} \log \sigma(-\tilde{v}_{w_i}^T v_{w_i}), \tag{4.4}$$

donde v_w y $\tilde{v}_w$ son representaciones de entrada y salida de la palabra w y S_k es un muestreo aleatorio de k palabras que constituyen los casos corruptos.

Estas técnicas permiten construir representaciones de calidad eficientemente. Debido a su relativamente reducido costo computacional es posible utilizarlos en corpus de texto de miles de millones de palabras.

Mikolov et al. (2013a), además de presentar *skip-gram* y *cbow*, observan que las representaciones sostienen relaciones sintácticas y semánticas. Por ejemplo, $w_{caminar} - w_{caminando} = w_{correr} - w_{corriendo}$ es una analogía sintáctica y $w_{montevideo} - w_{uruguay} = w_{atenas} - w_{grecia}$ semántica. Esta propiedad que los vectores mantienen da lugar a definir pruebas de evaluación basadas en las analogías. La evaluación de analogías habitualmente usada consiste en el porcentaje de aciertos en un conjunto de cuaternas de la forma a es a b como c es a d. La respuesta se considera correcta si d es la palabra más cercana a $w_b - w_a + w_c$. Levy y Goldberg (2014b) proponen una variante multiplicativa (en vez de aditiva) con buenos resultados.

4.3.4 GloVe

GloVe (por Global Vectors), introducido por Pennington et al. (2014), es un método para construir representaciones de las palabras formulando un problema de mínimos cuadrados a partir de las coocurrencias de palabras en un corpus. El método considera una ventana de tamaño fijo de palabras. El tamaño y dirección de la ventana son hiperparámetros del modelo.

La cantidad de coocurrencias de pares de palabras es almacenada en una matriz X, donde la entrada X_{ij} corresponde al número de veces que la palabra j ocurre en el contexto de i. Se define la probabilidad de que j ocurra en el contexto de i como

$$P_{ij} = P(j|i) = \frac{X_{ij}}{\sum_k X_{ik}} \tag{4.5}$$

lo que da lugar a una de las consideraciones centrales del método, el comportamiento de $\frac{P_{ik}}{P_{jk}}$ que puede ser resumido como:

$$\frac{P_{ik}}{P_{jk}} \begin{cases} > 1 & k \text{ más vinculado a } i \text{ que a } j \\ < 1 & k \text{ más vinculado a } j \text{ que a } i \\ \approx 1 & \text{en otro caso} \end{cases}$$

Se considera:

$$w_i^T \tilde{w}_k = \log P_{ik} \tag{4.6}$$

donde w_i y $\tilde{w}_k$ son representaciones constituidas por parámetros a aprender. Esta definición permite considerar

$$(w_i - w_j)^T \tilde{w}_k = \log \frac{P_{ik}}{P_{jk}}, \tag{4.7}$$

que relaciona a la diferencia de vectores con el cociente de probabilidades anteriormente mencionado.

De las ecuaciones (4.5) y (4.6), tenemos que

$$w_i^T \tilde{w}_k = \log X_{ik} - \log \sum_j X_{ij}, \tag{4.8}$$

donde $\log \sum_j X_{ij}$ es independiente de k, por lo tanto puede ser absorbido por b_i y para mantener la simetría se agrega $\tilde{b}_k$ resultando en

$$w_i^T \tilde{w}_k + b_i + \tilde{b}_k = \log X_{ik}. \tag{4.9}$$

Finalmente, la ecuación (4.9) es formulada como el siguiente problema de mínimos cuadrados

$$J = \sum_{i,k=1}^{V} f(X_{ik})(w_i^T \tilde{w}_k + b_i + \tilde{b}_k - \log X_{ik})^2,$$

donde V es el tamaño del vocabulario y f una función de peso con propiedades convenientes, como desvanecerse en 0 y no asignar pesos prioritariamente altos a las palabras raras o demasiado frecuentes.

Notar que de la resolución de este problema de mínimos cuadrados surgen dos representaciones para cada palabra (w_i y $\tilde{w}_i$). La representación retornada es $w_i + \tilde{w}_i$ porque experimentalmente conduce a mejores resultados.

4.4 Comparación entre modelos

La explosión y diversidad de métodos para construir representaciones distribuidas de las palabras revela que información estadística en los textos puede llevar a inferir, al menos parcialmente, el significado de las palabras. Es importante comparar y relacionar los modelos para aprender más de ellos y mejorarlos.

La distinción entre modelos de conteo y de predicción da lugar a una comparación en ese sentido. Baroni et al. (2014) presentan una comparación sistemática entre modelos de conteo y predicción que contempla distintas alternativas de medidas estadísticas y factorizaciones para los modelos de conteo. Para los modelos de predicción los experimentos se centraron en distintas configuraciones de *cbow*. Se realiza un extenso conjunto de evaluaciones donde se observa un desempeño significativamente mayor en los modelos de predicción.

Pennington et al. (2014) realizan experimentos de similitud y analogías de palabras para representaciones obtenidas por métodos de conteo, predicción y *GloVe*. Se contemplan los modelos de conteo *PMI-SVD* y *Hellinger PCA* (Lebret y Lebret, 2013). Para el caso de predicción, se consideran modelos como *skip-gram*, *cbow* y *GloVe*. La comparación muestra nuevamente un desempeño mejor en los modelos de predicción.

A pesar de que las comparaciones realizadas por Baroni y Pennington posicionan como más adecuados a los modelos de predicción frente a los de conteo, el estudio realizado por Levy et al. (2015) muestra que esto se debe a que los modelos de conteo carecen de ciertas consideraciones, que son las que brindan representaciones de mejor calidad en los modelos de predicción. En lo que sigue de la sección vamos a profundizar en este resultado.

4.4.1 Skip-gram como una factorización de la matriz de PMIs

Los consistentes resultados posicionaban a los modelos de predicción por encima de los de conteo en la construcción de representaciones distribuidas de las palabras hasta que Levy y Goldberg (2014c), muestra que *skip-gram* con negative sampling (SGNS) converge a una factorización de la matriz de *PMI*s desplazada por una constante global. Un resultado similar es probado para otro modelo de predicción mostrando que podría tratarse de un resultado más general y no un caso particular para SGNS.

El resultado para SGNS se formula reescribiendo la función objetivo del modelo, por cada par palabra-contexto y asumiendo una distribución uniforme de unigramas del vocabulario para el muestreo negativo, como

$$\#(w,c)\log\sigma(w.c) + k.\#(w).\frac{\#(c)}{|D|}.\log\sigma(-w.c),$$

donde k es la cantidad de ejemplos negativos y $\#(c)$ es la cantidad de ocurrencias del contexto c. Para optimizar la función objetivo se considera $x = w.c$ y se calculan las derivadas parciales respecto a x. Se busca el valor de x que vuelve cero a las derivadas y se tiene:

$$w.c = \log\frac{\#(w,c).|D|}{\#(w)\#(c)} - \log k,$$

donde el primer término corresponde a la *PMI* entre w y c y el segundo es una constante global que depende únicamente de la cantidad de ejemplo negativos.

Este resultado orienta la construcción de la medida *shifted positive PMI (SPPMI)*, con la cual se realizan pruebas, incluyendo además, reducciones de dimensión con valores singulares. La comparación muestra que, en las pruebas de similitud la medida creada y las reducciones superan a *SGNS*. Por otro lado, en las pruebas de analogías sintácticas, *SGNS* supera significativamente al resto. Los autores conjeturan que esto se debe a que *SGNS* le da prioridad a los pares más frecuentes, al contrario del resto que le da el mismo peso a todas las configuraciones de pares de palabras.

4.4.2 Transferencia de Hiperparámetros

El hecho de que *SGNS* sea una factorización de la matriz de PPMI, conecta al modelo con los modelos de conteo. Esto permite, además de tener mayor conocimiento del modelo, transferir consideraciones que este realice a los modelos de conteo llegando a resultado comparables (Levy et al., 2015).

Se distinguen tres tipos de hiperparámetros: *pre-procesamiento*, *métrica de asociación* y *post-procesamiento*. Los hiperparámetros de pre-procesamiento presentados son ventana contexto dinámico, *subsampling* y borrado de palabras raras. La métrica de asociación considerada es *shifted PPMI*, y se considera para el pos-procesamiento, la adición de los vectores de contexto al los de palabra, factorizaciones con una variante de la descomposición de valores singulares y normalización de los vectores resultado.

Los autores conducen una serie de pruebas con diferentes configuraciones de hiperparámetros y observan que es posible obtener resultados similares a SGNS con modelos de

conteo según las consideraciones especificadas. No obstante, en las pruebas de analogías sintácticas *SGNS* arroja los mejores resultados.

Capítulo 5

Representaciones Distribuidas para el Español

En este capítulo se presenta la construcción y evaluación de representaciones vectoriales para el español. Se adaptan conjuntos de evaluación del inglés al español y se evalúan las representaciones obtenidas. El capítulo concluye con un estudio cualitativo de las representaciones, mostrando que las representaciones generan de modo espontáneo clases léxicas del dominio de la temporalidad con propiedades interesantes.

5.1 Construcción de las representaciones

A pesar del progreso en métodos para obtener representaciones vectoriales de las palabras y formas de evaluar las representaciones obtenidas, al momento de realizar los experimentos de esta tesis no se disponía de ningún recurso validado para el español. De hecho, el único recurso disponible era el de Al-Rfou et al. (2013), en el trabajo llamado *polyglot*, que construyen representaciones para más de cien lenguajes , entre ellos el español. Pero, en este trabajo no se evalúan explícitamente las representaciones construidas. Además, la única dimensión disponible para el espacio de representación es 64 y como se verá más adelante, considerando dimensiones mayores pueden obtenerse mejores resultados[1].

Por lo anterior, se construyeron repertorios de representaciones vectoriales de las palabras y se da para ellas una evaluación inicial. Para evaluar la calidad de las representaciones construidas se consideraron pruebas de similitud y pruebas de analogías, adaptando para ambos casos conjuntos de evaluación en inglés. Se comentan ambos tipos de pruebas y los conjuntos utilizados en la sección 5.1.2. Los resultados obtenidos son inferiores a los

[1]El efecto de la dimensión considerada puede ser dependiente del método utilizado para construir las representaciones.

reportados en inglés pero esto es esperable debido a la diferencia de tamaño en los conjuntos de texto utilizados. De todas formas, se considera que los resultados son alentadores.

En simultáneo al desarrollo de los experimentos presentados en esta tesis se llevó a cabo un proyecto para la construcción de vectores en español, en el que se realizó la construcción de un corpus de apróximadamente seis mil millones de palabras para entrenar las representaciones (Azzinnari y Martínez, 2016). Se incluyen resultados de la evaluación explícita de vectores resultantes de este proyecto, corresponden a vectores de dimensión 300 utilizando *skip-gram* sobre la totalidad del corpus.

5.1.1 Construcción

Las representaciones fueron construidas utilizando la Wikipedia en español [2] y *GloVe*. El corpus fue procesado para quitar metadatos y etiquetas *XML*. Para esto se utilizó el *script* realizado por Matt Mahoney[3] modificado para considerar los caracteres con acento utilizados en español. Se utilizó el corpus en minúsculas, perdiendo la distinción entre nombres propios pero evitando cualquier error que pueda ser introducido por desambiguar las mayúsculas de comienzo de oración. El corpus final está constituido por 130 millones de palabras aproximadamente.

En algunos trabajos al construir representaciones vectoriales de las palabras, se sustituyen los valores numéricos por términos especiales indicando únicamente la cantidad de cifras del número (ej. 1815 $\rightarrow$ dddd). Esto puede tener un efecto positivo en la calidad de las representaciones aunque no se conocen resultados que lo afirmen. Sin embargo, en este trabajo se pretende utilizar las representaciones en el tratamiento de la temporalidad y los valores numéricos ocurren con frecuencia para referir a cantidades, años, días del mes, etc. Por este motivo, en las representaciones construidas se mantuvieron los valores numéricos y se estudia para los mismos el comportamiento de las representaciones obtenidas.

El entrenamiento fue realizado utilizando la implementación en *C* disponible en el sitio de *GloVe*[4]. El tiempo de entrenamiento para los experimentos mas costosos fue aproximadamente de 20 horas de procesamiento.

5.1.2 Evaluación

Para evaluar las representaciones construidas se consideran principalmente las pruebas de similitud y de analogías de relaciones entre las palabras al igual que Pennington et al. (2014),

[2] `http://dumps.wikimedia.org/eswiki/20150228/eswiki-20150228-pages-articles.xml.bz2`
[3] `http://mattmahoney.net/dc/textdata.html`
[4] `http://nlp.stanford.edu/projects/glove/`

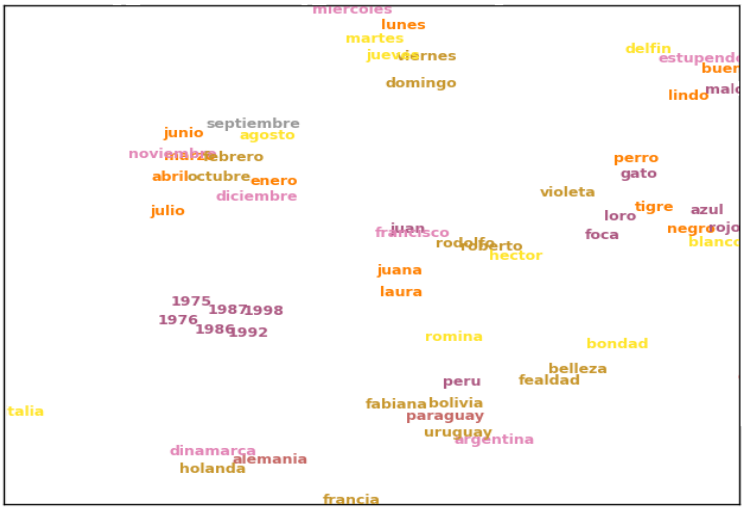

Fig. 5.1 Visualización de representaciones de palabras. La dimensión es reducida con t-sne, vectores de dimensión 150 son considerados. Notar la formación de agrupaciones con las palabras relacionadas.

pero antes de proceder con las pruebas se presentan alternativas útiles para observar la calidad de las representaciones basadas en visualizaciones.

Independientemente de contar con mecanismos sistemáticos para evaluar la calidad de las representaciones, es útil visualizar, al menos parcialmente las representaciones obtenidas. Al ser un espacio de alta dimensión la visualización no es una tarea trivial. Una alternativa es observar las palabras más próximas a una dada en término de las representaciones. También es útil comparar las distancias de una palabra a otras dos, verificando si las representaciones se comportan según los esperado. Esta es la esencia de la tarea de evaluación por similitud de palabras que se presenta más adelante. Estas alternativas, a pesar de ser claras, son centradas en cada palabra y es preciso contar con muchos casos para tener un panorama global de la representación.

Otra alternativa para visualizar las representaciones es reducir la dimensión para representarlas gráficamente. Se probaron varias técnicas para esto, considerando a *t-sne* (van der Maaten y Hinton, 2008) entre las que se obtuvieron las mejores visualizaciones. En la figura 5.1 se muestra una visualización de las representaciones obtenidas de dimensión 150, donde es posible apreciar la formación de agrupaciones de palabras relacionadas.

Similitud

Para evaluar la similitud de palabras se consideran conjuntos de datos formados por pares de palabras seguido de un valor que indica el grado de relacionamiento entre ellas.

Es necesario notar que al menos dos medidas pueden ser consideradas; por un lado la similitud de palabras indica el grado en que dos palabras significan lo mismo (ej. *silla* y *asiento* tendrían un valor alto), por otro lado, el grado de relacionamiento indica la vinculación entre las palabras sin que necesariamente tengan que denotar el mismo concepto. Por ejemplo, la palabra *silla* y *mesa* tendrían un grado alto de relacionamiento mientras que su grado de similitud es relativamente bajo. En los conjuntos de evaluación utilizados, salvo que se diga lo contrario, se considera el grado de relacionamiento de las palabras.

Para la tarea se consideraron las traducciones al español de los conjuntos *WordSim-353* (Finkelstein et al., 2002) y MC30 (Miller y Charles, 1991) provistas por Hassan y Mihalcea (2009) con correcciones de acentos. Además se tradujo manualmente el conjunto SimLex-999 (Hill et al., 2014) teniendo en cuenta los valores de cada par para desambiguar los casos con múltiples traducciones posibles.

Dim	**WS353**	**MC30**	**SL999a**	**SL999**
25	19.9	64.6	14.7	11.7
50	26.7	67.6	18.8	16.0
100	28.8	67.0	23.7	19.3
150	30.5	65.5	25.5	20.0
200	30.5	64.2	26.0	20.7
250	30.5	61.6	27.2	21.3
300-nabu-6b	54.7	71.4	32.4	22.3
200-en-6b	55.22	66.56	36.91	34.03

Tabla 5.1 Resultados de la tarea de similitud usando la medida de correlación de rangos de Spearman en varios conjuntos de evaluación. Se reportan resultados para varias dimensiones. Notar que SimLex-999 (SL999) mide similitud en vez de asociación. SL999a es el conjunto SimLex-999 considerando el valor de Assoc(USF) que refleja asociación en vez de similitud.

En la tabla 5.1 se presentan los resultados de la evaluación en los conjuntos de evaluación considerados. La medida utilizada es la correlación de rangos de Spearman. Los resultados obtenidos son significativamente más bajos que en inglés. Esto puede deber principalmente a que se utiliza un corpus de tamaño mucho menor. Además, sería apropiado considerar conjuntos construidos originalmente para español.

Analogías

Mikolov et al. (2013c) observan una relación entre las representaciones de la palabras con resultados sorprendentes. Al considerar dos pares de palabras bajo la misma relación (ej. *hablar* es a *habló* como *decir* es a *dijo*), el vector que va desde la representación de *hablar* a la representación de *habló* tiende a coincidir con el vector que va desde la representación de *decir* a

Dataset	**Dimension**						
	25	**50**	**100**	**150**	**200**	**300***	**200****
semantic							
capital-comm	40.4	65.1	72.5	74.4	75.4	84.4	98.6
capital-world	21.3	40.3	51.3	53.2	51.8	80.2	98.2
city-in-state	25.6	42.8	52.6	57.1	59.0	29.4	67.9
currency	0.3	0.7	0.7	0.7	0.6	4.7	27.6
family	62.6	78.0	79.6	81.8	80.1	92.5	93.5
syntactic	**25**	**50**	**100**	**150**	**200**	**300***	**200****
adj-to-adv	4.5	6.0	8.9	9.7	8.3	42.1	41.0
opposite	4.0	7.6	8.5	10.1	11.7	30.3	34.4
comparative	-	-	-	-	-	-	94.0
superlative	-	-	-	-	-	-	86.6
present-part	21.9	29.0	37.1	35.7	32.9	84.4	84.1
nation-adj	44.0	68.3	81.8	86.0	86.6	92.8	95.1
past-tense	12.3	21.4	26.9	27.5	27.7	77.5	81.8
plural	13.5	22.7	30.9	33.0	36.5	73.0	91.5
plural-verbs	26.9	39.8	47.5	45.7	43.1	87.1	83.7

Tabla 5.2 Resultados en la tarea de analogías de palabras para cada conjunto de evaluación y diferentes dimensiones. Los conjuntos son agrupados en sintácticos y semánticos manteniendo su clasificación original. Las analogías de comparativos y superlativos no aplican al español. $300*$ corresponde a los vectores de Azzinnari y Martínez (2016) y $200**$ a vectores provistos por Pennington et al. (2014) para el inglés construidos utilizando un corpus de seis mil millones de palabras.

la representación de *dijo*. La notación utilizada es (*hablar*:*habló*::*decir*:*dijo*) y la relación que sostienen las representaciones es $v_{hablar} - v_{habló} \approx v_{decir} - v_{dijo}$ donde v_w es la representación de la palabra *w*. Por lo tanto, si consideramos la pregunta *hablar*:*habló*::*decir*:*?*, podríamos responderla con la palabra cuya representación es la más próxima a $v_{decir} + (v_{habló} - v_{hablar})$. El test de analogías consiste en responder correctamente preguntas de este tipo.

Es posible considerar diversos tipos de relaciones, por ejemplo, país y ciudad capital, verbo y gerundio, género en relaciones familiares, entre otras. La relaciones pueden ser de carácter sintáctico o semántico. Las relaciones sintácticas son las basadas en una propiedad gramatical como infinitivo y gerundio (ej. *vivir*:*viviendo*) mientras que las semánticas en una propiedad como el país y la moneda que utiliza (ej. *uruguay*:*pesos*).

Se tradujo el conjunto de evaluación de analogías utilizado por Pennington et al. (2014), compuesto de 20.000 preguntas para evaluar la representación. En la tabla 5.2 se presenta el resultado de la evaluación de las representaciones en distintas dimensiones. El resultado es el porcentaje de preguntas resueltas correctamente, considerando que una respuesta es correcta si la palabra resultado se encuentra entre las cinco más próximas al vector computado.

5.2 Comportamiento de los términos temporales

Para estudiar el comportamiento de las representaciones distribuidas en los términos del léxico temporal, debido a la alta dimensionalidad del espacio, se consideraron distancias entre las palabras y técnicas de reducción de la dimensión.

5.2.1 Agrupamiento

La representaciones son capaces de identificar conjuntos de términos temporales como los nombres de los días de la semana, meses, etc. y permitir la generalización por términos relacionados. En la figura 5.2 se visualizan las representaciones de una selección de términos temporales al realizar una reducción de dimensionalidad a 2 con *t-sne* para representarlo gráficamente. En la representación se puede observar la tendencia a la formación de grupos de palabras relacionadas. Se agrupan los días de la semana, los meses, años, adverbios y números bajos habitualmente usados en los días del mes.

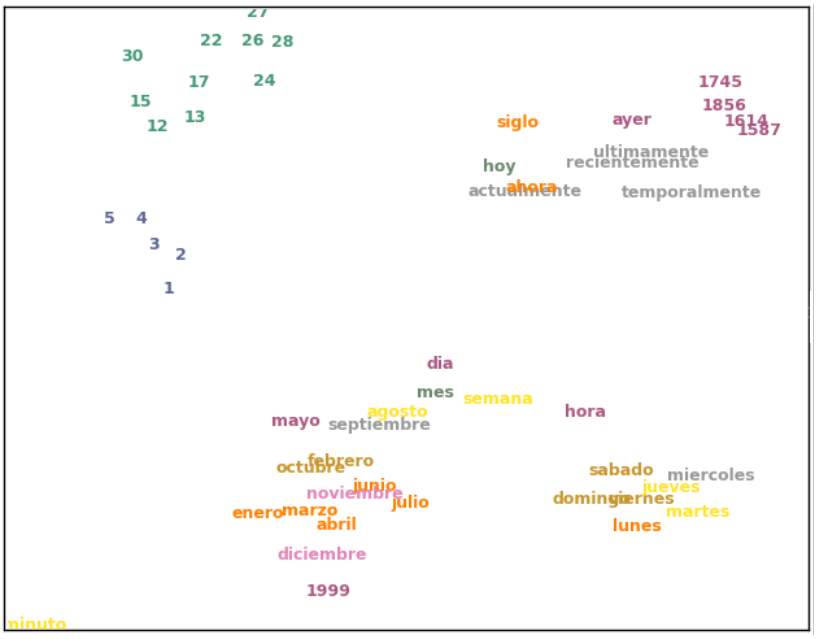

Fig. 5.2 Representación en 2 dimensiones usando *t-sne* de representaciones de dimensión 200 de algunos términos del léxico temporal.

Además de los días de la semana y meses es interesante observar el comportamiento de palabras como *amanecer* que especifica un momento del día. Otras palabras que denotan momentos del día como *atardecer* y *medianoche* tienen representaciones cercanas. Algo similar pasa con los nombres de los periodos como *neolítico*, los ejemplos mencionados se encuentran en la tabla 5.3.

Lunes	Enero	Amanecer	Neolítico
viernes	marzo	atardecer	paleolítico
jueves	febrero	mañana	mesolítico
miércoles	junio	noche	calcolítico
martes	abril	día	neolítico
sábado	diciembre	medianoche	datan
mañana	noviembre	anochece	pleistoceno
madrugada	octubre	mediodía	precerámico
sábados	agosto	ocaso	epipaleolítico
domingo	septiembre	madrugada	bronce
...	...	...	...

Tabla 5.3 Tabla con los términos más próximos ordenados por distancia de las representaciones de términos temporales.

El comportamiento de otras palabras con inclinación temporal también produce resultados alentadores. Por ejemplo, las palabras *inicio* y *final* cercanas a la palabra *comienzo*. Notar también que adverbios como *repentinamente* tienen próximo a *prematuramente* o *tempranamente*. Estos ejemplos se presentan en la tabla 5.4.

Comienzo	Antes	Repentinamente	Apresuradamente
inicio	después	súbitamente	marchar
dio	tras	muere	replegarse
antes	ya	falleció	precipitadamente
final	días	murió	desecaba
dando	luego	prematuramente	mudarse
llegada	ese	trágicamente	periódicamente
finales	meses	tempranamente	dirigiera
principio	tiempo	...	...
momento	comenzar		
...	...		

Tabla 5.4 Tabla con los términos más próximos ordenados por distancia de las representaciones de términos temporales.

Este comportamiento es adecuado para la detección supervisada de las expresiones. Mediante la agrupación de términos relacionados es posible generalizar ejemplos del conjunto de entrenamiento a casos no presentes. Esto impacta directamente en la cobertura del sistema.

5.2.2 Granularidad y Orden

En los ejemplos presentados anteriormente (sección 5.2.1), además de la formación de grupos, se puede observar una tendencia a preservar el orden correlativo de palabras como el nombre de los días de la semana, meses, etc. Por ejemplo, la palabra *miércoles* ocurre próximo a *martes* y *jueves* (ver tabla 5.3 y figura 5.2). Incluso, al reducir la dimensión a 1 utilizando *Principal Component Analysis* se reconstruyó el orden completo de los días de la semana.

Esta tendencia a que las representaciones contengan información del orden de los términos, posiblemente se deba a la existencia de secuencias de términos ordenados en el conjunto de texto de donde se infieren las representaciones. Esta propiedad se observa además con otros grupos de palabras con un orden establecido, como las cantidades numéricas y términos ordinales.

En las representaciones de los números, además de la tendencia a preservar el orden, los términos de una misma granularidad tienden a estar cerca. Por granularidad se entiende a la unidad menos significativa distinta de cero. Por ejemplo, 1502 tiene una granularidad de unidades, 1610 de decenas, 1400 de siglos y 1000 de milenios.

Si se considera la representación de un número del orden de las centenas, por ejemplo *1700*, otros números del orden de las centenas como *1600* o *1800* son próximos (notar además que son los números más próximos a *1700* en escala de centenas). Análogamente, con números del orden de las decenas como *1850*, están cerca números como *1840*, *1860* y *1870*. Sin embargo, al considerar un número como *1853*, números como *1855* y *1854* son los cercanos (ver tabla 5.5). Respecto a los términos ordinales, al considerar términos como *primero*, *segundo* y *tercero*, el término *primero* está próximo a términos como *segundo* pero también a *último*. La representación del término *segundo* está cerca de términos como *tercero*, *cuarto*, *quinto*, etc. Por otro lado, próximo al término *vigésimo* están términos como *décimo* y *trigésimo*.

En la figura 5.3 se presenta una reducción de dimensionalidad de las representaciones de términos numéricos que especifican años. Es posible apreciar agrupaciones con criterios como los mencionados anteriormente y la tendencia al ordenamiento de las secuencias de números. Es interesante observar como la secuencia *1920*, *1930*, ..., *1970*, *1980*, *1990* "conecta" el grupo de *1888*, *1889*, etc. con el grupo de *1991*, *1992*, etc.

Estos fenómenos que ocurren con el orden de los términos y la granularidad de cantidades numéricas son potencialmente útiles en la interpretación de las expresiones. Resulta interesante que estas propiedades se deriven únicamente de los contextos donde ocurren las palabras. Esto da la perspectiva de la posibilidad de interpretar expresiones, o al menos cierto tipo de expresiones, de forma supervisada sin la necesidad de especificar esta información explícitamente. Incluso puede dar lugar al ordenamiento no supervisado de expresiones.

Primero	Segundo	Vigésimo	1853	1850	1700	1999
luego	tercer	trigésimo	1855	1840	1600	1998
segundo	primer	décimo	1854	1849	1800	1995
mismo	cuarto	cuadragésimo	1856	1870	1500	1997
último	último	noveno	1852	1860	1400	1996
primer	quinto	quincuagésimo	1851	1880	1200	2002
posteriormente	primero	octavo	1865	1851	1100	2003
después	tercero	quinto	1849	1830	1300	1994
ese	sexto	séptima	1857	1890	2500	1993
otro	ese	sexagésimo	1859	1855	800	2004
...	...	...	...	...	...	...

Tabla 5.5 Tabla con los términos más próximos ordenados por distancia de las representaciones de términos ordinales y años.

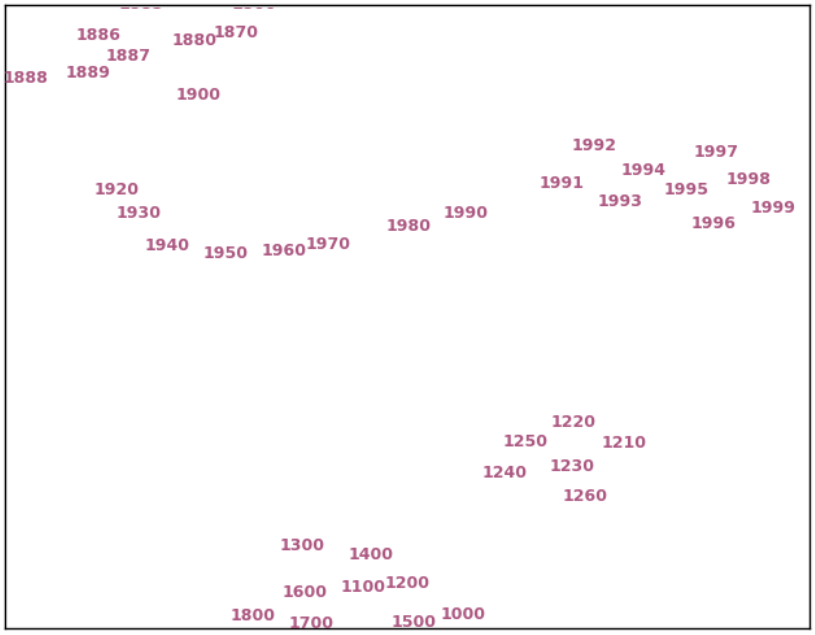

Fig. 5.3 Representación en 2 dimensiones usando *t-sne* de representaciones de dimensión 200 de números.

Capítulo 6

Detección y Clasificación de Expresiones Temporales con Redes Neuronales

En el capítulo anterior se presentó la construcción de representaciones vectoriales para las palabras en español. Estas representaciones dieron resultados interesantes respecto al léxico de la temporalidad (sección 5.2). En este capítulo se proponen modelos neuronales, que usan las representaciones construidas, para reconocer y clasificar las expresiones temporales que ocurren en un texto.

Se estudia la capacidad de las representaciones y los modelos propuestos para tratar la temporalidad. No se pretende mejorar los resultados de los modelos mediante técnicas que incorporen conocimiento específico del dominio de la temporalidad. En los modelos planteados, no se realiza ningún análisis léxico o sintáctico del texto ni se usa ningún recurso externo como ontologías o clases de palabras. Los únicos mecanismos de generalización usados son las representaciones vectoriales de las palabras inferidas de sus contextos y el entrenamiento supervisado de los modelos neuronales. Los modelos no incluyen ninguna información explícita del dominio ni se realiza ningún procesamiento previo o posterior a la aplicación del modelo que lo haga.

La principal complejidad que presenta el uso de modelo neuronales reside en la diversidad de decisiones de diseño que deben ser tomadas, dicho de otro modo, en determinar cuál es el modelo da los mejores resultados. Estas decisiones incluyen la representación de la entrada, funciones de activación, cantidad y tamaño de capas ocultas, tipo de capas, forma y parámetros de entrenamiento , técnicas de regularización y función objetivo, entre otras. Se entrenan y evalúan decenas de modelos explorando sistemáticamente distintas configuraciones de parámetros y decisiones de diseño. Para abordar la diversidad de configuraciones posibles,

se dividen los experimentos en grupos intentando aislar las decisiones más relevantes que deben ser tomadas.

El resto del capítulo se estructura como sigue. Primero, se plantea la tarea como un problema de etiquetado de secuencias. Luego se introducen los modelos de redes neuronales considerados. Se entrenan y evalúan diversos modelos guiados por las principales decisiones de diseño que deben ser tomadas. Se analiza el comportamiento de distintas variantes y técnicas de regularización. Se concluye con una discusión de los mejores modelos, experimentos para el inglés y comparaciones con el estado del arte.

6.1 Detección y clasificación como etiquetado de secuencias

Para detectar y clasificar expresiones temporales se plantea el problema como una tarea de etiquetado de secuencias, precisamente de *clasificación de segmentos* usando la nomenclatura utilizada en (Graves, 2012), una referencia ampliamente usada en este trabajo.

A partir de una secuencia de entrada (texto escrito en el que se quiere detectar y clasificar las expresiones temporales), se le asigna a cada una de sus palabras una etiqueta que indica el rol de la palabra en términos de las expresiones temporales del texto de entrada. Para asignar la etiqueta de cada palabra, se considera su representación vectorial, y las representaciones de palabras previas o posteriores, es decir, información del contexto.

Las etiquetas utilizadas corresponden al esquema *BILOU*, el cual ha mostrado ser apropiados en tareas de extracción de entidades con nombre (Ratinov y Roth, 2009). El esquema especifica para cada palabra si es el comienzo (**B**eginning), interior (**I**nside) o fin (**L**ast) de una expresión, si es una expresión unitaria (**U**nit) o si se trata de una palabra que no forma parte de una expresión temporal (**O**utside). En la tabla 6.1 puede verse la descripción y ejemplos de cada etiqueta.

	Nombre	**Descripción**	**Ejemplo**
B	Begin	Comienzo de una expresión.	Estaremos$_{O}$ 20$_{\mathbf{B}}$ días$_{L}$ de$_{O}$ vacaciones$_{O}$.$_{O}$
I	Inside	Interior de una expresión. [1]	Ellos llegaron$_{O}$ un$_{B}$ rato$_{\mathbf{I}}$ antes$_{L}$.$_{O}$
L	Last	Final de una expresión.	Eso$_{O}$ ocurrió$_{O}$ el$_{B}$ pasado$_{I}$ sábado$_{\mathbf{L}}$.$_{O}$
O	Out	Exterior. No hay expresión.	Aguarde$_{\mathbf{O}}$ un$_{B}$ minuto$_{L}$ por$_{\mathbf{O}}$ favor$_{\mathbf{O}}$.$_{\mathbf{O}}$
U	Unit	Expresión unitaria. [2]	Hoy$_{\mathbf{U}}$ la$_{O}$ tierra$_{O}$ no$_{O}$ es$_{O}$ la$_{O}$ misma$_{O}$.$_{O}$

Tabla 6.1 Descripción de las etiquetas del sistema *BILOU*.

El esquema presentado permite anotar la extensión de las expresiones que ocurren en un texto, esto constituye la tareas de detección de expresiones. Para la tarea de clasificación, se adapta el esquema anterior para especificar el tipo de cada expresión. La alternativa directa es replicar el conjunto completo de etiquetas para cada tipo de expresión con excepción de la etiqueta O. Se tendría el repertorio de etiquetas:

$$\{O\} \bigcup_{i \in T} \{B_i, I_i, L_i, U_i\}, \qquad T = \{duration, set, date, time\},$$

donde T es el conjunto de tipos de expresiones considerado, constituido por la clasificación presentada en la sección 2.1.1.

Esta forma de adaptar el esquema original para soportar, además de la detección, la clasificación de las expresiones, tiene la desventaja de aumentar considerablemente la cantidad de etiquetas y admitir una cantidad exponencialmente mayor de configuraciones inválidas. Por ejemplo, expresiones donde sus palabras son clasificadas con tipos diferentes. En adición a lo anterior, es frecuente que expresiones de distinto tipo tengan varias palabras en común. Incluso, expresiones idénticas (en su forma de escribirse) pueden tener tipos (y significados) diferentes según el contexto.

Para evitar estos inconvenientes, se distingue únicamente la última palabra de la expresión, teniendo una etiqueta $\mathbf{L}_i$ y $\mathbf{U}_i$ por cada tipo de expresión y manteniendo una única versión de las restantes etiquetas del esquema. El beneficio de distinguir únicamente las etiquetas L y U en contraste de distinguir también a B e I es cuestionable y requiere pruebas pero no se profundiza en este aspecto.

En la tabla 6.2 se presenta el conjunto completo de etiquetas utilizado para la clasificación de expresiones.

Tipo	**Etiquetas**
fecha (da)	L_d, U_d
momento del día (t)	L_t, U_t
duración (du)	L_{du}, U_{du}
conjunto (s)	L_s, U_s
otras	B, I, O

Tabla 6.2 Esquema de etiquetas utilizado para la clasificación de expresiones temporales.

Definir las tareas de detección y clasificación de esta manera resulta adecuado para encarar el problema con técnicas de aprendizaje automático y en particular con modelos de redes neuronales. En la siguiente sección se presentan los modelos considerados, detalles de entrenamiento e implementación.

6.2 Modelos Propuestos

Los modelos propuestos para resolver la detección y clasificación de expresiones temporales clasifican las palabras de un texto según el esquema *BILOU*, presentado en la sección anterior (6.1). Los modelos utilizan como entrada los conjuntos de representaciones vectoriales de las palabras previamente construidos (sección 5.1). Una introducción a los modelos neuronales considerados puede encontrarse en el capítulo 3.

Los modelos considerados clasifican una palabra a partir de su representación vectorial y una representación del contexto en un texto de entrada. La aplicación del modelo sobre todas las palabras del texto resuelve la tarea de detección o detección y clasificación, según sea el esquema considerado.

Cada activación del modelo (*forward pass*) retorna la etiqueta de la palabra donde fue aplicado. El orden de aplicación del modelo es relevante para los modelos recurrentes o que consideren explícitamente en su entrada la salida de la activación en palabras previas o posteriores. En el caso de los modelos *feedforward* que únicamente consideran las representaciones de la palabra a clasificar y otras de contexto, el orden de aplicación es irrelevante. Sin embargo, se considera por defecto el orden de aplicación secuencial de izquierda a derecha (fig. 6.1) debido a su analogía con la forma habitual de recepción de los lenguajes.

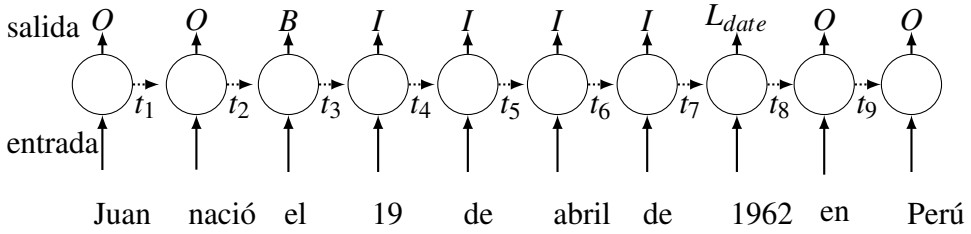

Fig. 6.1 Ejemplo de aplicación secuencial de un modelo.

El contexto brinda información valiosa en el tratamiento de las expresiones temporales. Se consideran dos formas de incluir información del contexto lingüístico en los modelos: el contexto denominado *ventana* y el contexto brindado por los modelos recurrentes.

En lo que sigue de la sección se presenta la organización de los modelos considerados y las distintas formas de incluir información del contexto.

6.2.1 Estructura general

Los modelos considerados están organizados en capas. La capa de entrada recibe la representación de la palabra a clasificar. La información es procesada y transmitida, pasando

por las capas ocultas hacia la capa de salida (fig. 6.2). Si el funcionamiento de la red es adecuado, cada capa transforma la representación que recibe a otra más adecuada para inferir la respuesta.

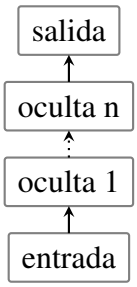

Fig. 6.2 Modelo en capas.

Todos los modelos presentados utilizan especificaciones similares de capa de entrada y salida. Las variaciones entre modelos se dan en el tipo y cantidad de capas ocultas, junto con su respectiva forma de entrenamiento, información contextual y técnicas de regularización.

Como ya se mencionó y se muestra más adelante en los resultados obtenidos, la información contextual es de gran importancia para esta tarea. Para la detección y clasificación de expresiones temporales las palabras cercanas a la expresión, su contexto lingüístico, brindan información necesaria, por lo cual incluir información del contexto en el modelo impacta significativamente en los resultados. Más aún, en la interpretación de las expresiones temporales, es indispensable contar con información contextual referente a la temporalidad, como ser, fecha de creación del documento, momento del habla e interpretación de otras expresiones temporales y eventos necesarios para la interpretación de expresiones relativas, ya sea anafórica o deíctica. Nos centramos en la detección y clasificación con la consideración del contexto lingüístico formado por las palabras cercanas, dejando del lado por el momento, la consideración de otros tipos de información contextual.

En la siguiente sección se presenta el contexto de tipo ventana, una forma de consideración de contexto lingüístico que es aplicable a todos los modelos presentados. Luego, se presenta la consideración de información contextual con el contexto provisto por los modelos recurrentes, en particular *LSTM*s bidireccionales.

6.2.2 Contexto Ventana

Las redes *feedforward* no aportan inherentemente información del contexto en la clasificación. Sin embargo, es posible incluir información contextual explícitamente en la entrada. Por ejemplo, las representaciones de las palabras cercanas a la palabra a clasificar.

Se define como contexto de tipo *ventana* a la concatenación de las c_i representaciones anteriores a la palabra, para el caso de contexto *izquierdo* o *anterior*, y de las c_d palabras posteriores para el contexto *derecho* o *posterior*. Cuando se consideran igual cantidad de palabras de contexto anterior y posterior, se denomina contexto ventana simétrico.

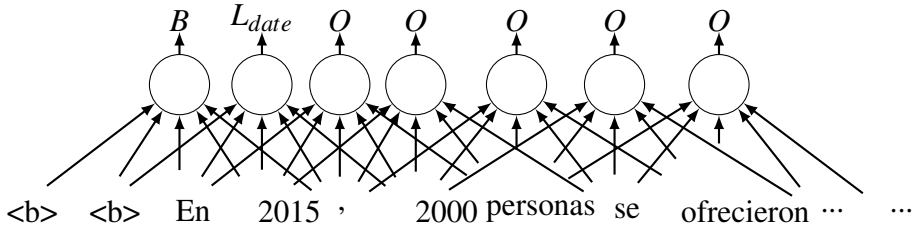

Fig. 6.3 Aplicación de modelo con una ventana simétrica de dos palabras de contexto.

Por ejemplo, si se considera un contexto ventana izquierdo de tamaño dos y derecho de tamaño tres, la entrada recibida por la red neuronal sería $x = [w_{i-2}w_{i-1}w_iw_{i+1}w_{i+2}w_{i+3}]$, el vector concatenación de las representaciones de las palabras del texto de entrada, donde i es la palabra a clasificar. En la figura 6.3 se presenta un diagrama ejemplificando un contexto simétrico de tamaño dos. En el comienzo y fin de oración se utilizan palabras de relleno, en todos los casos se consideró el vector nulo.

Este tipo de contexto tiene la desventaja de tener un alcance rígido indistintamente al caso presentado ante el modelo. Además, no es apropiado para la consideración de contextos grandes (ej. 20 palabras). Sin embargo, puede resultar altamente efectivo en la consideración del contexto de corto alcance.

Al incrementar la cantidad de palabras de contexto de tipo ventana se incrementa el tamaño de la entrada del modelo. Una práctica habitual es considerar el tamaño de las capas ocultas en relación al tamaño de la capa anterior y posterior. Por lo tanto, un cambio en el tamaño de la ventana considerada debe ser acompañado con un ajuste del tamaño de todas las capas ocultas del modelo.

6.2.3 Contexto de recurrencia

Los modelos recurrentes considerados están basados en la retroalimentación de las capas ocultas en si mismas. Es decir, la salida de una capa oculta (recurrente) forma parte de la entrada de la misma capa en la siguiente activación (fig. 6.4). La memoria de las activaciones previas provista por las capas recurrentes brinda información contextual a cada activación del modelo. En la sección 3.3 se encuentra una descripción de los modelos neuronales recurrentes.

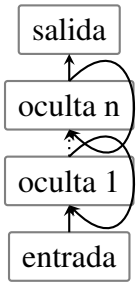

Fig. 6.4 Modelo recurrente en capas.

En estos modelos la ejecución y entrenamiento del modelo debe ser necesariamente secuencial. El sentido de ejecución se corresponde con el tipo de contexto considerado. Si se considera en sentido directo (izquierda a derecha), corresponde al contexto anterior. Por el contrario, el sentido inverso corresponde al contexto posterior de la palabra.

Para incluir información de ambos contextos (izquierdo y derecho), se consideran modelos bidireccionales, en particular *Bidirectional Long Short-Term Memory* (BLSTM) Graves y Schmidhuber (2005), debido al éxito de los *LSTMs*.

6.3 Análisis de los Modelos

La diversidad de configuraciones que permiten los modelos neuronales es una desventaja al momento de usarlos en una tarea concreta. Sobre todo si no se tiene resultados previos que oficien como guía. Resulta impracticable explorar exhaustivamente el espacio de posibles configuraciones.

Para hacer frente a esta situación, se realizan experimentos en función a cierta característica del modelo o técnica para la que resulta de interés estudiar el comportamiento. En los experimentos se pretende observar el comportamiento de distintas variantes de forma independiente, por lo tanto se adapta el resto del ambiente para que la comparación revele el efecto de la variante estudiada sin ser afectada por consideraciones adicionales.

Una desventaja del tratamiento aislado de las variantes es que no se estudia la interacción de configuraciones. No se realiza un estudio de relación de hiperparámetros y técnicas, debido a la cantidad de modelos que deberían entrenarse y su costo computacional. En (Greff et al., 2015) se observa que hay independencia entre hiperparámetros básicos[3] con búsquedas aleatorias, pero estos resultados no son directamente aplicable en este caso.

[3]Hiperparámetros básicos como tamaño de capas ocultas y tasa de aprendizaje.

Se observa la interacción en conjunto de técnicas que muestran ser efectivas por separado para investigar su convivencia. Se observa mediante resultados experimentales si sus beneficios se potencian, se absorben o se contrarrestan.

El entrenamiento y evaluaciones se realizó con los datos para el español impartidos en la competencia TempEval-3 (UzZaman et al., 2012). Como los datos de evaluación no están disponibles, se fraccionaron los datos ofrecidos para entrenar en un conjunto de entrenamiento y otro de evaluación. El conjunto de entrenamiento cuenta con **878** expresiones temporales y el conjunto de evaluación con **216**, en la tabla 6.3 se muestra información de los corpus utilizados. Se usaron las representaciones de palabras inferidas de la Wikipedia usando GloVe (Pennington et al., 2014) presentadas en 5.1.

		Expresiones Temporales				
Nombre	**Palabras**	**Fecha**	**Duración**	**Hora**	**Conjunto**	**Total**
TEval13_es_train	46.687	585	215	49	29	878
TEval13_es_test	12.197	164	36	8	8	216

Tabla 6.3 Información del corpus de TempEval 2013 para el español.

El tamaño limitado del corpus es un contratiempo para los enfoques de aprendizaje supervisado en contraste a los que incluyen conocimiento mediante reglas. Sin embargo, es un buen escenario para poner a prueba la capacidad de generalización de las representaciones distribuidas de las palabras en situaciones de pocos datos. Otra desventaja es que el tamaño reducido del conjunto de evaluación afecta la calidad de la evaluación y disminuye el impacto de pequeñas variaciones en los resultados de los experimentos.

Todas los modelos fueron entrenados con *RMSprop* (Tieleman y Hinton, 2012), una variante de *backpropagation*. En todos los casos un valor de *momento* de 0.9 fue considerado. El criterio de parada es el prolongado del modelo, precisamente, si la mejora de la función objetivo no supera $1x10^{-5}$ por 30 etapas de entrenamiento. Por el tamaño limitado de los datos no se consideró un conjunto de validación para realizar *early stopping* en el entrenamiento. La implementación de los modelos se realizó con la librería *Theanets* (Johnson et al., 2015), basada en *Theano* (Theano Development Team, 2016), orientada a la manipulación de modelos neuronales.

La evaluación se realiza mediante precisión (*precision*) y cobertura (*recall*) a nivel de expresión de la salida de cada modelo evaluado respecto a los datos de evaluación. La medida de evaluación global se lleva a cabo mediante la habitual medida F con igual balance para ambas.

6.3.1 Dimensión de Palabras

La dimensión del espacio de representación de las palabras es un factor crucial. Desde un punto de vista neuronal puede verse como la cantidad de neuronas involucradas en la representación. Al considerar dimensiones grandes se tiende a las representaciones dispersas y en particular al caso localista donde cada palabra tiene asociada una neurona y es representada por la actividad en dicha neurona dejando las restantes en cero (*one-hot encoding*). Por el contrario, con dimensiones menores, se tiende a representaciones compactas basadas en patrones de actividad, cuyo extremo corresponde al caso donde todas las palabras son representadas con valores distintos en una única neurona.

Desde los enfoques iniciales de representación de las palabras se ha observado la arbitrariedad de la dimensión del espacio. No se tiene un entendimiento profundo del impacto de la dimensión en la calidad de la representación y no hay resultados teóricos que guíen la elección de la dimensión.

Algunos trabajos muestran que incrementar la dimensión considerada para el espacio de representación presenta mejoras en la calidad de los vectores (Mikolov et al., 2013a; Pennington et al., 2014). Esta tendencia a mejorar no parece ilimitada pero no se conocen resultados al respecto y no ha sido objeto de estudio en esta tesis. Por otro lado, los vectores compactos tienen características deseables como los beneficios en cuanto a costo computacional.

Para observar el impacto de la dimensión en la tarea de detección consideramos representaciones de varias dimensiones en similares condiciones. Se entrenaron modelos manteniendo el resto del ambiente los más invariante posible.

Los modelos usados fueron *feedforward* de tres capas con tres palabras de contexto simétrico. Para mantener la proporción entre capas se ajustó el tamaño de la capa oculta a tres veces la dimensión de las representaciones de las palabras. Se observó un incremento continuo en la cobertura pero la precisión comienza a degradarse. Se muestran los resultados en la tabla 6.4.

La capa oculta recibe una ventana de siete palabras centrada en la palabra a clasificar que transformará a un vector del cual se infiere el resultado. El tamaño de la capa oculta, especifica la dimensión del espacio de ese vector de representación intermedia. Se observa el comportamiento de reducir a la mitad el tamaño de la capa oculta para los modelos considerados que dieron mejores resultados (dimensión de palabra 150 y 200). En ambos casos se observó una ligera mejora en los resultados, aportada principalmente por un incremento en la precisión (tabla 6.6). Este resultado muestra que un aumento en la dimensión del espacio de representación, no necesariamente lleva a mejores resultados.

Dim	Tiempo	Train Acc	P	R	F
25	3063.90s	1x10-6	72.97	50.00	59.34
50	5658.37s	1x10-6	76.14	62.04	68.37
100	11512.65s	396x10-4	**82.35**	64.81	72.54
150	19051.10s	396x10-4	80.79	66.20	72.77
200	33794.77s	396x10-4	78.61	**68.06**	**72.95**

Tabla 6.4 Comparación de resultados de detección en ambientes similares variando la dimensión de representación de las palabras. Se usaron modelos *feedforward* con una capa oculta y tres palabras de contexto simétrico.

Dim	Hid	P	R	F
150	450	80.79	66.20	72.77
	225	**82.28**	66.67	**73.66**
200	600	78.61	68.06	72.95
	300	79.14	**68.52**	73.45

Tabla 6.5 Resultados de detección al reducir la dimensión de la capa oculta a la mitad de los modelos mas significativos de la tabla 6.4.

Este comportamiento no se mantuvo para la clasificación. Se estudió el comportamiento del mismo experimento cambiando la capa de salida y entrenado para la clasificación de expresiones. En este caso, reducir la capa oculta, no significó en una mejora global sino en una leve degradación. Ni siquiera se observa correlación entre los valores de precisión y cobertura. En la tabla 6.6 se presentan los resultados obtenidos. Es posible que este cambio de comportamiento tras reducir la capa oculta se deba al cambio en la dimensión y distribución de los datos de salida pero son muy pocos datos para obtener mayores conclusiones al respecto.

Dim	Hid	P	R	F
150	450	**77.46**	62.04	**68.89**
	225	75.00	**62.50**	68.18
200	600	77.97	**63.89**	**70.22**
	300	**78.61**	62.96	69.92

Tabla 6.6 Resultados de la clasificación de expresiones con los mejores previamente usados en detección 6.5 pero cambiando la capa de salida para la clasificación. Notar como cambia el efecto de reducir la capa oculta respecto al resultado detección.

Los modelos que resuelven la clasificación de expresiones realizan también la tarea de detección. Por la tanto interesa saber de que manera afecta la clasificación a la detección. En

la tabla 6.7 se compararon resultados de detección, entre los modelos que dieron mejores resultados para la detección y el mismo modelo cambiando la capa de salida para la clasificación. En general la precisión de la detección fue mejorada en los modelos que además clasifican a la expresión. Esto puede deberse a que la información de la clasificación es utilizada en la detección. Puede verse en uno de los casos que la clasificación no mejora la precisión de la detección. Dicho caso corresponde al modelo con menor cantidad de parámetros, y una cantidad insuficiente de parámetros podría ocasionar una situación en la que considerar la tarea adicional de clasificar las expresiones sature al modelo. Se requieren más experimentos para avalar esta afirmación u obtener otras conclusiones.

Dim	Hid	P	R	F	
150	450	82.66 (80.79)	66.20 (66.20)	73.52 (72.77)	+0.75
	225	80.00 (82.28)	66.67 (66.67)	72.73 (73.66)	-0.93
200	600	83.05 (78.61)	**68.06 (68.06)**	**74.81 (72.95)**	+1.86
	300	**83.81 (79.14)**	67.13 (68.52)	74.55 (73.45)	+1.10

Tabla 6.7 Comparación de resultados en la tarea de detección en modelos entrenados para clasificación con el modelo de iguales características pero que únicamente realiza la detección (resultado entre paréntesis).

En cuanto a la generalización de palabras a casos no vistos, se observó un caso positivo con la palabra *semestre*, que en los datos de entrenamiento no ocurre y en los datos de test dos veces y algunos de los modelos son capaces de detectar al menos una ocurrencia [4]. Esto muestra que el modelo fue capaz de generalizar a partir de palabras como *trimestre* y *cuatrimestre* que ocurren en los datos de entrenamiento a expresiones con palabras que no ocurren, como *semestre*. Esto es debido a la similitud de las representaciones entre las palabras *trimestre*, *cuatrimestre* y *semestre*. La información no supervisada provista por las representaciones de las palabras, manipulada mediante modelos neuronales, permite la generalización de datos supervisados a casos que contengan palabras no contempladas durante el entrenamiento.

6.3.2 Contexto

Tanto para la extracción como para la clasificación de expresiones temporales, las palabras cercanas al elemento (su contexto lingüístico) contienen información necesaria para realizar la tarea. Basta con pensar algún caso numérico, sin información de las palabras previas o

[4] Por ejemplo dimensión de palabra 200, tres palabras de contexto simétrico y una capa oculta de tamaño 300.

posteriores, en muchos casos no es posible determinar si se trata de una expresión temporal y menos aún clasificarla.

Como se vio en la sección 6.2.2, el contexto de tipo ventana consiste en usar como entrada la concatenación de las representaciones de una ventana de palabras que incluye la palabra a clasificar. Se experimentó incrementando la cantidad de palabras de contexto izquierdo (sin contexto derecho) de modelos *feedforward* con una capa oculta. El tamaño de la capa oculta esta definido en función del largo del contexto. Se puede ver que si bien el contexto es un factor crucial, arrojando mejores resultados en todos los casos en relación a la versión sin contexto, contextos grandes perjudican los resultados. En la tabla 6.8 se muestran los resultados para la clasificación de expresiones.

Izq	Hid	P	R	F
0	100	60.64	26.39	36.77
1	100	67.31	32.41	43.75
2	200	**69.83**	**37.50**	**48.79**
3	300	67.00	31.02	42.40
4	400	69.81	34.26	45.96

Tabla 6.8 Comparación de los resultados de clasificación al incrementar el contexto izquierdo sin considerar contexto derecho. El modelo es *feedforward* de dimensión de palabra 200 y el tamaño de la capa oculta se calcula en relación al tamaño del contexto.

En la tabla 6.9 se presentan los resultados de realizar el experimento análogo para el contexto derecho. Se observó un comportamiento similar en sentido relativo. Es interesante notar que en todos los casos su uso mejora considerablemente los resultados respecto a los obtenidos para el contexto izquierdo. Los resultados muestran que el contexto derecho brinda más información para esta tarea. Si este fenómeno continúa en la representación de otras estructuras y otras lenguas, puede mostrar características de los lenguajes y la comunicación vinculadas a nuestro sistema de atención. Disponer de la parte más informativa en el contexto derecho (o futuro) a la unidad procesada, sugiere que la información de mayor relevancia para el receptor aún no ha sido recibida, impulsándolo a retener su atención.

La consideración de contexto simétrico arrojó resultados considerablemente mejores que ambos casos de contexto unidireccional. Al igual que en los casos anteriores, se encontró el mejor resultado al considerar dos palabras de contexto simétrico, pero es interesante notar que el mejor resultado de cobertura se obtuvo al considerar tres palabras de contexto simétrico, con medida F próxima al mejor resultado.

En la siguiente sección se muestra el desempeño de contextos neuronales en la tarea. El entrenamiento de modelos recurrentes es mucho mas costoso computacionalmente que el entrenamiento de modelos *feedforward*. Sin embargo, los modelos recurrentes ofrecen una

Der	Hid	P	R	F
0	100	60.64	26.39	36.77
1	100	**64.97**	47.22	54.69
2	200	63.64	**48.61**	**55.12**
3	300	62.42	45.37	52.55
4	400	62.42	43.06	50.96

Tabla 6.9 Experimento análogo al presentado en la tabla 6.8 pero para el contexto derecho. Los mejores resultados se obtienen al considerar 2 palabras de contexto (al igual que ocurrió con el contexto izquierdo).

Bid	Hid	P	R	F
0	100	60.64	26.39	36.77
1	100	72.78	56.94	63.90
2	200	**80.84**	62.50	**70.50**
3	300	76.92	**64.81**	70.35
4	400	76.86	43.06	55.19

Tabla 6.10 Experimento con iguales condiciones al realizado para el contexto izquierdo (tabla 6.8) y derecho (tabla 6.9) pero con contexto simétrico.

consideración más flexible de la información del contexto y a su vez no excluyente al que ofrecen los contexto de tipo ventana.

Contexto Neuronal

Una alternativa no excluyente al contexto de tipo ventana es el contexto considerado por los modelos recurrentes. Mediante retroalimentaciones en la capa oculta, consideran activaciones previamente realizadas, permitiendo con la aplicación secuencial de la red, considerar el contexto correspondiente a la dirección en la que fue aplicado.

Las consideraciones contextuales de los modelos recurrente son de mayor flexibilidad que la información explícita provista por el contexto de tipo ventana, sin embargo , su interpretación es más compleja y en los experimentos realizados no presentó buenos resultados sin la consideración adicional de contexto de tipo ventana.

Como los resultados mejoran significativamente al considerar ambos contextos, anterior y posterior, se consideran modelos bidireccionales, particularmente, modelos *Bidirectional Long-Short Term Memory* (BLSTM) Graves y Schmidhuber (2005).

En los experimentos realizados con modelos *BLSTM*, los resultados obtenidos fueron muy inferiores a los de los modelos *feedforward* con contexto de tipo ventana. En la tabla

h	Steps	P	R	F
150	100	54.20	32.87	40.92
200	100	60.75	30.09	40.25
600	100	63.16	**33.33**	**43.64**
300	15	64.36	30.09	41.25
600	15	64.44	27.31	38.43
1000	15	**71.44**	30.56	42.85
2000	15	71.25	26.39	38.51
600	3	64.00	14.09	24.06

Tabla 6.11 Resultados en clasificación de expresiones con modelos *BLSTM* sobre palabras de dimensión 200 sin contexto de tipo ventana. Los modelos tienen una única capa oculta con diferentes tamaños.

6.11 se presentan los resultados. Se consideraron distintos tamaños para la capa recurrente y profundidades 100, 15 y 3 para la recurrencia.

6.3.3 Tamaño de Capas Ocultas

Las capas ocultas corresponden a una parte fundamental del modelo, encargándose de transformar la representación recibida en una intermedia para resolver la tarea. El tamaño de las capas ocultas refiere a la dimensión del espacio de representación intermedia. Capas ocultas grandes van a tender a sobreajustarse a los datos de entrenamiento mientras que capas demasiado pequeñas no van a ser capaces de resolver la tarea adecuadamente.

En la sección 6.3.1 se observó que la dimensión de representación de las palabras y el tamaño de las capas ocultas impacta en los resultados totales. No se conocen resultados que orienten el tamaño de las capas ocultas.

Se entrenó una secuencia de modelos para la clasificación de expresiones sobre palabras de dimensión 200 con tres palabras de contexto anterior y posterior variando el tamaño de la capa oculta (tabla 6.12). Los mejores resultados se obtuvieron al considerar un tamaño de capa oculta a partir de 300, es decir, la cuarta parte del tamaño de representación de contexto total (600 contexto anterior y 600 posterior).

6.3.4 Ruido

Cuando el modelo se ajusta a los datos de entrenamiento pero no es capaz de generalizar a nuevos datos se denomina *sobreajuste*. El sobreajuste es habitual cuando el modelo tiene demasiados parámetros para los datos de entrenamiento. El modelo memoriza los datos de entrenamiento sin ser capaz de generalizar correctamente a otros casos.

Hid	P	R	F
100	74.58	61.11	67.18
200	75.28	62.04	68.02
300	78.61	62.96	69.92
400	77.27	62.96	69.39
500	76.95	63.42	69.54
600	79.21	**65.27**	**71.57**
700	**79.31**	63.89	70.77
1000	76.40	62.96	69.04
2000	79.19	63.43	70.44

Tabla 6.12 Resultados en clasificación de expresiones de modelos *feedforward* sobre palabras de dimensión 200, con tres palabras de contexto simétrico variando el tamaño de la capa oculta.

Realizar pequeñas perturbaciones a la entrada o representaciones intermedias durante el entrenamiento puede evitar el sobreajuste y llevar a mejorar la cobertura del modelo. Se aplicaron perturbaciones, de acuerdo a una distribución *gaussiana* centrada en cero y usando distintas varianzas, a los datos de entrada y representaciones intermedias. La varianza se interpreta como un parámetro para graduar el nivel de ruido aplicado.

Incluir ruido en la entrada resultó en una mejora considerable en cobertura y ligera en precisión para valores de ruido a partir de 0.1 (tabla 6.13). Para valores de ruido menores a 0.1 los resultados fueron levemente inferiores, como los conjuntos de datos son pequeños quizás se deba a particularidades de la instancia de entrenamiento. Los mejores resultados se obtuvieron al considerar 0.2 de varianza del ruido inyectado degradándose al considerar 0.3.

$\textbf{Ruido}_I$	P	R	F
0.00	78.61	62.96	69.92
0.01	77.20	61.11	68.22
0.05	77.40	63.43	69.72
0.10	79.33	65.74	71.90
0.20	**80.66**	**67.59**	**73.55**
0.30	78.65	64.81	71.06

Tabla 6.13 Resultados en clasificación de expresiones de modelos *feedforward* sobre palabras de dimensión 200, con tres palabras de contexto simétrico y capa oculta de tamaño 300, con distintos niveles de ruido en la entrada.

Se realizó el experimento análogo, inyectando ruido en la capa oculta. En la tabla 6.14 se presentan los resultados obtenidos. Los resultados presentan una ligera mejora en cobertura acompañada de una pérdida de precisión, significando globalmente en una mejora menos pronunciada respecto a aplicar ruido a la entrada.

Ruido$_H$	**Dim**	**P**	**R**	**F**
0.00	300	78.61	62.96	69.92
0.01	300	**79.41**	62.50	69.95
0.05	300	77.60	**65.74**	**71.18**
0.10	300	77.40	63.43	69.72
0.20	300	77.01	62.04	68.72

Tabla 6.14 Resultados en clasificación de expresiones de modelos *feedforward* sobre palabras de dimensión 200, con tres palabras de contexto simétrico y capa oculta de tamaño 300, con distintos niveles de ruido en la capa oculta.

Se entrenó un modelo con las mejores niveles de ruido en la entrada y en la capa oculta para analizar la interacción de ambas técnicas. Ambos efectos no se favorecieron conjuntamente, obteniéndose resultados similares a los obtenidos sin considerar ruido e incluso degradando la precisión en casi 2%.

En cuanto a la detección de expresiones, al igual que en la clasificación, se obtuvo el mejor resultado al considerar un valor de 0.2 de ruido en la entrada y sin ruido en la capa oculta. Los mejores resultados obtenidos en la detección fueron de 78.59 de medida F para el caso estricto y 81.42 para el caso relajado, donde se admiten corrimientos de una palabra en la extensión de la expresión detectada.

6.3.5 Dropout

La técnica de *dropout* consiste en aleatoriamente volver cero algunas entradas de la red o capas intermedias. Puede verse como una forma de ruido donde se anula completamente algunas componentes dejando intactas otras. También es conocido con el nombre de máscara multiplicativa. Se considera como parámetro de *dropout* a la fracción de unidades de la representación que son llevadas a cero.

Se experimentó con distintos valores de *dropout* para la salida de la capa oculta, de un modelo *feedforward* sobre palabras de dimensión 200, con tres palabras de contexto simétrico y capa oculta de tamaño 300. En la tabla 6.15 se puede observar que un efecto positivo con valores bajos degradándose la cobertura gradualmente al considerar valores mayores a 0.01 de *dropout* .

En cuanto a la detección, también se obtuvo el mejor resultado con 0.01 de valor de *dropout* en la capa oculta siendo 75.70 de medida F en el caso estricto y 81.84 para el caso relajado.

Dropout	P	R	F
0.00	78.61	62.96	69.92
0.001	78.03	62.50	69.41
0.01	**80.57**	**65.28**	**72.12**
0.05	78.29	63.43	70.08
0.10	75.82	63.89	69.35
0.20	77.14	62.50	69.05

Tabla 6.15 Resultados en clasificación de expresiones de modelos *feedforward* sobre palabras de dimensión 200, con tres palabras de contexto simétrico y capa oculta de tamaño 300, con distintos niveles de *dropout* en la capa oculta.

6.3.6 Regularizaciones L1 y L2

El sobreajuste puede reflejarse con valores altos en los parámetros aprendidos. Entrenar el modelo evitando valores altos en los parámetros puede llevar a evitar el sobre ajuste. Esta técnica se denomina *weight decay*.

Al agregar en la función objetivo el término $\lambda_{L2}\|\theta\|_2$, donde $\|.\|_2$ es la norma L2 y θ el vector de parámetros a aprender del modelo, da el efecto de reducir la magnitud de los parámetros considerando además de la función a optimizar. El factor λ_{L2} indica el grado de exigencia de valores bajos en los parámetros.

Se entrenó el mismo modelo base, considerado anteriormente, con distintos valores para la regularización L2. El modelo entrenado tiene una capa oculta de dimensión 300, 3 palabras de contexto simétrico sobre representaciones de las palabras de dimensión 200 (tabla 6.16). Se puede observar una mejora en los resultados al considerar $\lambda_{L2} = 0.001$. En ningún caso la técnica perjudica los resultados.

L2	P	R	F
0.00	78.61	62.96	69.92
0.0001	78.21	64.81	70.89
0.001	80.11	65.28	**71.94**
0.01	77.58	62.50	69.23
0.05	79.21	**65.43**	71.57
0.10	**81.21**	62.04	70.34
0.20	78.82	62.04	69.43

Tabla 6.16 Resultados en clasificación de expresiones de modelos *feedforward* sobre palabras de dimensión 200, con tres palabras de contexto simétrico y capa oculta de tamaño 300, con distintos niveles de *L2* .

Por otro lado, además de la tendencia a valores bajos en los parámetros aprendidos, puede haber un efecto positivo la presencia de valores dispersos en las representaciones, es decir con algunos valores en cero. Esto es inspirado por patrones dispersos en la corteza visual de los mamíferos. Una técnica para tender a parámetros dispersos es incluir a la función objetivo el término $\lambda_{L1}\|\theta\|_1$, donde $\|.\|_1$ es la norma *L1*. Incluir este término hace que se reduzcan a cero algunas componente si no perjudica al resultado.

L1	P	R	F
0.00	78.61	62.96	69.92
0.0001	74.71	60.18	66.67
0.001	**80.00**	62.96	70.47
0.01	78.89	**65.74**	**71.72**
0.05	74.85	59.26	66.15
0.10	74.30	61.57	67.34
0.20	77.21	56.50	65.24

Tabla 6.17 Resultados en clasificación de expresiones de modelos *feedforward* sobre palabras de dimensión 200, con tres palabras de contexto simétrico y capa oculta de tamaño 300, con distintos niveles de *L1*.

Análogamente a los experimentos con la regularización L2, se analiza el efecto de considerar distintos valores para λ_{L1}. En la tabla 6.17 están los resultados obtenidos en la detección y clasificación de expresiones. Puede observarse que valores superiores a 0.05 de regularización L1, degradan significativamente los resultados. El mejor resultado se obtuvo con 0.01 que presentó una mejora de 2 puntos en la medida F.

6.3.7 Cantidad de capas

La cantidad de capas ocultas es un punto central en la definición de modelos neuronales. Es interesante notar que los modelos sin capas ocultas no pueden resolver adecuadamente algunos problemas. Un ejemplo clásico es el caso de la función lógica *XOR* que no puede ser modelada sin capas ocultas.

Las redes con varias capas ocultan presentan dificultades en su entrenamiento. Las primeras capas ocultas, las cercanas a la entrada, no son debidamente ajustadas por *backpropagation*. Casos exitosos de entrenar redes profundas se encontraron en las redes convolutivas, inspiradas en la corteza visual. Estos modelos pueden ser entrenados adecuadamente para profundidades mayores que los modelos *feedforward* clásicos.

Técnicas de pre-entrenamiento, entrenando de forma no supervisada las capas desde la entrada a la salida, antes del entrenamiento supervisado con *backpropagation* han presentado

una mejora importante en los resultados respecto a la versión inicializada aleatoriamente. Una amplia referencia a los modelos profundos y aplicaciones puede encontrarse en el libro *Learning Deep Architectures for AI* de Yoshua Bengio (2009).

h1	h2	h3	h4	P	R	F
300	-	-	-	78.61	62.96	69.92
300	100	-	-	**79.67**	**67.13**	**72.86**
300	200	100	-	74.07	64.81	69.13
600	400	200	100	64.25	57.41	60.64

Tabla 6.18 Resultados en clasificación de expresiones de modelos *feedforward* sobre palabras de dimensión 200, con tres palabras de contexto simétrico variando la cantidad de capas ocultas.

Evaluar el efecto de la profundidad del modelo es una tarea difícil. Es importante notar que considerar modelos con más capas, invlucra la necesidad de definir particularidades de cada capa, principalmente tipo y tamaño. Para simplificar esta situación se consideraron distintas profundidades en modelos homogéneos, es decir, con todas las capas del mismo tipo. En cuanto al tamaño, se consideraron dimensiones escalonadas en las capas ocultas. Cada capa oculta transforma la entrada recibida a una de menor dimensión.

El primer experimento realizado, fue agregar al modelo utilizado como base en otras secciones, una capa oculta adicional entre la capa oculta ya existente y la capa de salida. Se consideró un tamaño 100 la capa oculta agregada. La inclusión de la capa adicional significó en una notoria mejora en los resultados, principalmente en cobertura (ver tabla 6.18).

Al observar el efecto favorable de considerar un modelo con dos capas ocultas (en comparación al modelo con una única capa oculta), se entrenaron y evaluaron modelos con tres y cuatro capas ocultas. En este caso los resultados muestran que considerar más de dos capas ocultas no tuvo un efecto positivo. Es importante notar, que distintar configuraciones en los tamaños de las capas ocultas puede cambiar esta situación. Al igual que considerar técnicas de pre-entrenamiento para inicializar el modelo.

h1	h2	h3	P	R	F
200	-	-	60.75	30.09	40.25
300	150	-	**68.42**	**54.17**	**60.46**
300	200	100	61.15	45.83	52.52

Tabla 6.19 Resultados en clasificación de expresiones con modelos *BLSTM* sobre palabras de dimensión 200 sin contexto de tipo ventana. Se evaluan modelos con distintas cantidades de capas ocultas recurrentes.

Respecto a los modelos recurrentes, aunque presentaron resultamos muy inferiores a los presentados por los modelos *feedforward* con una ventana de contexto, se estudia el efecto de considerar mas capas ocultas, también formadas por *BLSTMs*. Al igual que en el caso anterior, la consideración de una capa adicional significó una mejora en el comportamiento del modelo que también descendió al considerar más de dos capas ocultas (tabla 6.19). Se presentan los resultados al considerar una, dos y tres capas ocultas recurrentes. Estos modelos fueron entrenados considerando 100 pasos de recurrencia.

6.4 Discusión de los resultados

En las secciones anteriores del capítulo actual, se entrenaron y evaluaron distintas variantes de modelos neuronales, incluyendo modelos *feedforward* y recurrentes. Se probaron distintas configuraciones estructurales de los modelos y se evaluaron técnicas de regularización.

Una primera observación respecto a todos los experimentos realizados, es que la inicialización aleatoria de los pesos de los modelos puede llevar a que instancias distintas arrojen resultados diferentes. Para amortiguar esta situación, sería adecuado realizar varias instancias del mismo experimento y comparar resultados pero esto consumiría un tiempo de cómputo considerable. Es importante notar que cada experimento consume desde un par de horas a varios días. Esto realizando los experimentos en una computadora con unidad de procesamiento gráfico compatible con *CUDA*, de lo contrario, en una computadora estándar puede demorar semanas. Para lidiar con esta situación, se repitieron algunos experimentos en situaciones cuestionables y otros aleatoriamente, reportandose el mejor resultado obtenido. No se encontraron entre las distintas instancias de un mismo experimento diferencias drásticas.

De los experimentos realizados, es inmediato notar la importancia del contexto para resolver esta tarea. Tanto las ventanas de contexto, como la consideración implícita de los modelos recurrentes, presentaron mejoras importantes. Este comportamiento era de esperar. Sin embargo, el contexto de recurrencia provisto por los *BLSTM*s se comportó muy inferiormente que las consideraciones contextuales de ventana. A pesar de que se cree que estos resultados pueden mejorarse con un estudio en mayor profundidad de los modelos recurrentes, se hipotetiza que la radical diferencia entre ambos enfoques se debe a que la tarea resuelta se basa principalmente en el contexto cercano. Esto justificaría la ventaja de la consideración explícita de la cercanía de la palabra clasificada, provista por la ventana de contexto, frente a la flexibilidad del contexto de los modelos recurrentes.

Respecto a consideraciones estructurales del modelo, se presentó un efecto positivo al considerar dos capas ocultas, en lugar de una. Este comportamiento no se mantuvo para profundidades mayores. Es posible que técnicas de pre-entrenamiento mejoren el compor-

tamiento de modelos profundos (ej. 3 capas ocultas) pero no se realizaron experimentos al respecto. El tamaño de las capas ocultas presentó variaciones oscilantes para tamaños superiores a 300, degradándose para valores inferiores. Por lo tanto, no es posible obtener conclusiones más allá del efecto negativo de los tamaños inferiores.

Se consideraron las técnicas de regularización de ruido en la entrada y capa oculta, *dropout*, *L1* y *L2*. Aunque todas las técnicas consideradas presentaron mejoras respecto al modelo base, la mejora más significativa fue dada mediante la consideración de ruido en la entrada. Se cree que esto se debe principalmente a que el ruido en la entrada aumenta la capacidad de asociación del modelo entre palabras relacionadas (a partir de la información provista por las representaciones vectoriales de las palabras), mejorando considerablemente la cobertura del modelo. Es interesante notar, a diferencia de lo que ocurre habitualmente con técnicas de aprendizaje automático, que el modelo se adecua al ruido en la entrada, mejorando la cobertura sin perder precisión. Esto muestra un caso favorable de la robustez de los modelos neuronales. En la tabla 6.20 se presentan los mejores resultados obtenidos de cada familia de experimentos. Se recuerda que el modelo tomado como base es de 3 palabras de contexto simétrico de dimensión 200, con una capa oculta de dimensión 300.

h1	**h2**	**ruido$_I$**	**dropout**	**L1**	**L2**	**P**	**R**	**F**
300	-	-	-	-	-	78.61	62.96	69.92
600	-	-	-	-	-	79.21	65.27	71.57
300	100	-	-	-	-	79.67	67.13	72.86
300	**-**	**0.20**	**-**	**-**	**-**	**80.66**	**67.59**	**73.55**
300	-	-	0.01	-	-	80.57	65.28	72.12
300	-	-	-	0.01	-	78.89	65.74	71.72
300	-	-	-	-	0.001	80.11	65.28	71.94

Tabla 6.20 Resultado de los modelos más performantes en cada familia de experimentos.

Para observar el comportamiento de combinar distintas técnicas que mostraron dar buenos resultados, se consideraron experimentos con dos capas ocultas, ruido en la entrada y *dropout* simultáneamente. Respecto a estos dos últimos, puede observarse que la consideración conjunta, con valores que independientemente dieron buenos resultados, degrada la cobertura del modelo. Esta situación cambia al disminuir la cantidad de ruido inyectado. Como puede observarse en la tabla 6.21, la consideración de dos capas ocultas se comporta adecuadamente en conjunto con el ruido en la entrada y *dropout*. Al final de la tabla se incluyen resultados de modelos entrenados con los vectores de dimensión 300 provistos por Azzinnari y Martínez (2016) comentados y mostrando resultados de evaluaciones explícitas en la sección 5.1.

Aunque la consideración de ruido y *dropout* en un modelo de dos capas ocultas presentó resultados ligeramente superiores respecto al mejor modelo hasta el momento, esta mejora

h1	h2	ruido$_I$	dropout	P	R	F
300	-	0.20	-	80.66	67.59	73.55
700	-	0.2	-	81.03	65.28	72.31
700	400	0.2	-	**81.36**	66.67	73.28
400	150	0.2	0.1	80.35	64.35	71.46
350	120	0.1	0.1	80.32	**68.06**	**73.68**
600	-	0.1	-	81.21	**68.06**	**74.05**
450	200	0.1	-	**81.92**	67.13	73.79

Tabla 6.21 Resultados en la clasificación de expresiones combinando variantes que muestran efectos positivos independientemente.

fue respecto a la clasificación y no a la detección de expresiones, donde el mejor modelo hasta el momento sigue dando mejores resultados. Este resultado es superado por uno de los modelos entrenados utilizando los vectores de Azzinnari y Martínez (2016) (ver tabla 6.22).

h1	h2	ruido$_I$	dropout	P	R	F
300	-	0.2	-	86.19	**72.22**	**78.59**
700	-	0.2	-	**86.78**	69.91	77.44
700	400	0.2	-	85.88	70.37	77.35
350	120	0.1	0.1	84.70	71.76	77.69
600	-	0.1	-	**87.29**	**73.15**	**79.60**
450	200	0.1	-	86.44	70.83	77.86

Tabla 6.22 Resultados en la detección de expresiones para los modelos de la tabla 6.21.

Como conclusión se remarca la eficacia de inyectar ruido en las representaciones de las palabras utilizadas como entrada de los modelos neuronales para abordar la tarea en cuestión. También es de destacar el efecto positivo de considerar dos capas ocultas en lugar de una. Ambas consideraciones se comportan adecuadamente en conjunto, aunque sin presentar, por el momento, una mejora significativa respecto al modelo que únicamente considera ruido en la entrada.

6.4.1 Evaluación Cualitativa

El mejor resultado en cuanto al reconocimiento de las expresiones temporales se alcanzó con la consideración de 0.2 puntos de ruido en la entrada en el modelo con 3 palabras de contexto simétrico, con dimensión 200 en el espacio de representación de las palabras y 300 de capa oculta. Se considera este modelo para todos los experimento cualitativos realizados en esta sección.

Se muestra la capacidad de reconocimiento del modelo presentando una conjunto de casos donde se realiza el reconocimiento correctamente (tabla 6.23). En el caso 1 se puede observar la capacidad del modelo de distinguir entre un número que refiere a una fecha y otro que no. En el caso 2, el modelo es capaz de detectar correctamente una duración que involucra la palabra *últimos* y en el caso 3 una fecha completa especificada por día, mes y año. En los casos 4 y 5 se puede apreciar que el modelo detecta correctamente casos como *"hoy lunes"* y *"hoy, jueves"* admitiendo la opcionalidad de la coma. En el último de los casos presentados, el modelo clasifica correctamente la expresión *"los minutos finales"*. Es interesante notar que el modelo de detecta el comienzo, interior y fin de las expresiones reconocidas, sin brindarle información de la etiqueta previamente asignada, es decir, únicamente utilizando la información de las palabras anteriores y posteriores a la palabra clasificada.

$Entrada_1$	ha	ido	menguando	desde	1997	,	cuando	había	192	agentes	.
$Salida_1$	O	O	O	O	U_{da}	O	O	O	O	O	O
$Entrada_2$	revisiones	que	ha	tenido	en	los	últimos	años	,	algo	para
$Salida_2$	O	O	O	O	O	B	I	L_{du}	O	O	O
$Entrada_3$	fueron	detenidos	el	6	de	noviembre	de	1997	en	la	rochelle
$Salida_3$	O	O	B	I	I	I	I	L_da	O	O	O
$Entrada_4$	exportar	a	partir	de	hoy	lunes	productos	petroleros		a	
$Salida_4$	O	O	O	O	B	L_{da}	O	O		O	
$Entrada_5$	rumania	la	última	semana	,	tras	volver	hoy	,	jueves	
$Salida_5$	O	B	I	L_{da}	O	O	O	B	I	L_{da}	
$Entrada_6$	henry	)	en	los	minutos	finales	,	el	holandés	quedó	
$Salida_6$	O	O	O	B	I	L_du	O	O	O	O	

Tabla 6.23 Casos del conjunto de evaluación donde la salida fue la esperada.

En algunos casos, aunque no se obtuvo la salida esperada según el conjunto de evaluación, la incorrectitud del resultado obtenido es discutible. Esto puede deberse a errores en el conjunto de evaluación o sutilezas del esquema de anotación. En este último es esperable que el modelo se adapte a los criterios de anotación pero es interesante observar las diferencias presentadas por el modelo, pues podrian reflejar una forma con mayor consistencia al resto del esquema e impulsar futuros cambios en el esquema. En la tabla 6.24 se muestran los únicos tres casos encontrados con estas características. El caso 1 es un error de anotación, debido a que en otras partes del conjunto se encuentran casos similares anotados correctamente. Respecto al caso 2, no es un error de anotación, pero es discutible si debe ser considerado un error. La palabra *período* denota información temporal y forma parte de la expresión temporal *el período 2000 a 2006* en su totalidad pero TimeML tiene el criterio de anotar los intervalos expresados por sus dos extremos como dos expresiones separadas. En el tercer

caso el modelo no detecta la ocurrencia de la expresión *la nueva fecha* pero esta expresión podría considerarse como no temporal en este caso, debido al uso del pronombre *cuál* en lugar de *cuándo*.

Entrada_1	aterrizó	a	las	8	.	15	hora	local	(	07	.	15	gmt	)	,
Esperado_1	O	O	B	I	I	L_t	O	O	O	B	I	I	L_t	O	O
Salida_1	O	O	B	I	I	I	I	L_t	O	B	I	I	L_t	O	O

Entrada_2	68	.	000	millones	de	euros	para	el	período	2000	a	2006
Esperado_2	O	O	O	O	O	O	O	O	O	U_{da}	O	U_{da}
Salida_2	O	O	O	O	O	O	O	B	I	L_{da}	O	U_{da}

Entrada_3	nadie	sabe	cuál	es	la	nueva	fecha	que	propone
Esperado_3	O	O	O	O	B	I	L_{da}	O	O
Salida_3	O	O	O	O	O	O	O	O	O

Tabla 6.24 Casos del conjunto de evaluación donde la salida no fue la esperada pero la no correctitud es discutible.

Por otro lado, se analizaron casos donde el modelo dió un resultado incorrecto. En la tabla 6.25 se muestran algunos casos del conjunto de evaluación donde el modelo cometió errores en la detección o clasificación. En el caso 1 el modelo no fue capaz de incluir dentro de la expresión a la palabra *"navideña"*. Esta palabra no se encuentra en el conjunto de entrenamiento y el modelo no fue capaz de reconocerla a través de otras palabras con una representación cercana. En el caso 2 no fue capaz de detectar la palabra *"fines"* que forma parte de la expresión *"los fines de semana"*. La palabra *"fines"* tienen una única ocurrencia en el conjunto de entrenamiento (*"fines de 1999"*). En el caso 3, la expresión *"la madrugada de hoy sábado"* es detectada correctamente pero el modelo la clasifica como *date* en lugar de *time*. Notar que la palabra *"madrugada"* (que tiene una única ocurrencia en el conjunto de entrenamiento) es central para clasificar correctamente la expresión. Es interesante notar además que la palabra se encuentra a distancia 3 de la palabra que lleva la etiqueta con la información de la clasificación, esto implica que la información fue abarcada por la ventana de contexto considerada pero aún así la clasificación no fue correcta. En el caso 4 se muestra que el modelo comete errores detectando incorrectamente temperaturas y clasificándolas como expresiones temporales de tipo duración.

En los ejemplos anteriores se mostraron casos donde el modelo se comporta adecuadamente y algunos de los principales problemas en situaciones donde no se obtiene la solución correcta. En la siguiente sección se comparan cuantitativamente los resultados obtenidos con otros modelos, uno de los cuales constituye el actual estado del arte.

Entrada_1	los	protagonistas		se	conocen		una	tarde	navideña		en	unos	almacenes
Esperado_1	O	O		O	O		B	I	L_{du}		O	O	O
Salida_1	O	O		O	O		B	L_{du}	O		O	O	O
Entrada_2	el	déficit	es	especialmente			grave	los	fines	de	semana	,	cuando
Esperado_2	O	O	O	O			O	B	I	I	L_s	O	O
Salida_2	O	O	O	O			O	O	O	B	L_{da}	O	O
Entrada_3	agenda	programada		,	la	madrugada		de	hoy	sábado		registró	
Esperado_3	O	O		O	B	I		I	I	L_t		O	
Salida_3	O	O		O	B	I		I	I	L_{da}		O	
Entrada_4	temperatura			de	10	grados		bajo	cero	,		según	
Esperado_4	O			O	O	O		O	O	O		O	
Salida_4	O			O	B	L_{du}		O	O	O		O	

Tabla 6.25 Casos del conjunto de evaluación donde el modelo dió una respuesta incorrecta.

6.4.2 Comparación con otros trabajos

La comparación con otros trabajos no puede realizarse adecuadamente por no haber tenido acceso a los mismos datos de evaluación. De todos modos, se considera útil incluir al menos algún valor comparativo. Como se comentó en la sección 6.3, se fraccionó el conjunto de entrenamiento para disponer de un conjunto de evaluación. En definitiva, el modelo fue entrenado con una parte del conjunto de entrenamiento y evaluado con un conjunto de evaluación de características similares pero diferente.

La comparación se realiza en base al modelo que arrojó los mejores resultados en la detección de expresiones temporales. Como se vio en la sección 6.4 hubo un modelo que dio resultados ligeramente mejores en la clasificación e inferiores en la detección pero debido a que los resultados de otros modelos obtenidos refieren únicamente a la detección, no se consideró este último modelo para la comparación.

El modelo considerado tiene un única capa oculta de tamaño 300, con 3 palabras de contexto simétrico, dimensión 200 de palabras y un valor de 0.2 de varianza de perturbaciones realizadas a la entrada. En la tabla 6.26 se presentan la comparación de los resultados. Se incluye ademas los resultados con el modelo que dió mejores resultados utilizando los vectores de Azzinnari y Martínez (2016).

Para cada modelo se incluye el resultado de aplicar el modelo y posteriormente una heurística básica para corregir etiquetas inconsistentes (en la tabla 6.26 está indicado con un asterísco). La heurística consiste en corregir una etiqueta incorrecta en tres situaciones. Primero, si una *O* es seguida por *I*, se sustituye dicha *I* por *B* o U_{da} según sea el caso [5].

[5] Se consideró el tipo *date* por ser el más frecuente.

	P(r)	R(r)	F_1(r)	F_1(s)
HeidelTime	**96.0**	**84.9**	**90.1**	**85.3**
TIPSemB-F	93.7	81.9	87.4	82.6
FSS-TimEx	86.6	52.3	65.2	49.5
ANNTime	92.8	77.8	84.6	78.6
ANNTime*	91.2	81.5	86.1	79.2
ANNTime-nabu	91.7	76.8	83.6	79.6
ANNTime-nabu*	91.4	83.3	87.2	79.9

Tabla 6.26 Resultados de clasificación para el español de la competencia tempeval-3. ANNTime refiere a los mejores resultados obtenidos en este trabajo. ANNTime* refiere al mismo modelo aplicando la heurística para corregir etiquetas inconsistentes. ANNTime-nabu y ANNTime-nabu* corresponde a los mejores modelos utilizando los vectores de Azzinnari y Martínez (2016).

Segundo, si una *O* es precedida por *B* o *I* y seguida de *I* o *L*, se sustituye dicha etiqueta por *I*. Tercero y último, si una *I* está seguida de una *O*, se sustituye dicha *I* por L_{da} o U_{da} según sea el caso. La aplicación de la heurística mejoró los resultados en casi 4% respecto a la cobertura del modelo y degradó 1.6% en la precisión, resultando en una mejora global (F) de 1.5%, en el caso de solapamiento para los vectores básicos y una mejora de 3.6% para los vectores de dimensión 300.

6.4.3 Resultados para el Inglés

En esta sección se muestran los resultados de considerar modelos neuronales análogos pero para resolver la tarea en inglés, teniendo en cuenta las lecciones aprendidas a lo largo de los experimentos realizados para el español. Se utilizan los datos impartidos para el inglés en *TempEval 2013*, en la tabla 6.27 se muestran sus características. Se utilizan las representaciones impartidas por Pennington et al. (2014) de dimensión 200, construidas con un corpus de seis mil millones de palabras.

Nombre	Palabras	Fecha	Duración	Hora	Conjunto	Total
TEval13_en_silver	718.746	11.133	1.346	192	68	12.739
TEval13_en_platinum	7.003	96	34	4	4	138

Tabla 6.27 Información del corpus de TempEval 2013 para el inglés.

El enfoque adoptado, al no incorporar conocimiento especifico del idioma ni del dominio, se aplica directamente. El corpus de entrenamiento es veinte veces más grande que el utilizado en español, esto incrementa el tiempo de entrenamiento. Esta es la principal dificultad encontrada para realizar los experimentos para el inglés.

Se entrenó un modelo *feedforward*, con tres palabras de contexto simétrico, sobre representaciones de dimensión 200, con una única capa oculta de tamaño 300 y un nivel de 0.2 de ruido en la entrada, este fue el modelo que arrojó los mejores resultados en la detección para el español. Se entrenó el modelo durante 3.74 días sin alcanzar el criterio de parada. El modelo alcanzó un valor de 77.97% de medida F en la detección de expresiones.

h1	**h2**	**ruido**$_I$	**P**$_{det}$	**R**$_{det}$	**F**$_{det}$	**Acc**$_{class}$	**F**$_{class}$
300	-	0.20	93.88	66.67	77.97	93.47	72.88
300	100	-	95.83	66.67	78.63	**95.65**	75.21
900	200	-	**96.80**	65.94	78.45	95.60	75.00
600	100	0.20	95.88	**67.39**	**79.15**	93.54	74.04
Lee et al. (2014)			86.1	80.4	83.1	93.4	**85.4**

Tabla 6.28 Resultados de la detección y clasificación de expresiones para el inglés.

Luego se entrenó un modelo de dos capas ocultas, la más próxima a la capa de entrada de un tamaño de 300 y la otra de 100. Al igual que el modelo anterior se consideraron tres palabras de contexto simétrico, con representaciones de palabras de dimensión 200. La consideración de este modelo se basó en los resultados obtenidos para el español. Una observación de este modelo es que el error en el entrenamiento se reduce con mayor rapidez en comparación al caso comentado anteriormente. Luego de 1.86 días de entrenamiento se obtuvo un resultado de 78.63% de medida F en la detección. En la tabla 6.28 se detallan los resultados obtenidos.

Aumentar el tamaño de los *batches* de entrenamiento, debido al tamaño total del corpus, reduce el tiempo drásticamente. Se aumentó diez veces el tamaño del *batch* (de 64 a 640) y se entrenó un modelo con dos capas ocultas (tamaños 600 y 100) y un valor de 0.2 de ruido en la entrada. El modelo se entrenó durante 3.3 días, alcanzando el criterio de parada. Este modelo presentó una leve mejora en la cobertura de la detección respecto a los otros pero no mejoró los resultados obtenidos en la clasificación.

En cuanto a la clasificación de las expresiones, en todos los casos los modelos tuvieron un valor de medida F para la clasificación inferior al trabajo de Lee et al. (2014), esto es de esperar por que este valor esta influenciado por la detección. En cuanto al *accuracy*, en todos los casos se tuvieron mejores resultados[6] pero es preciso notar que este valor es afectado positivamente por el resultado inferior en la detección.

[6]Esta medida fue calculada mediante $AttrAccuracy = AttrF1/EntityExtractionF1$ (UzZaman et al., 2012)

Capítulo 7

Conclusiones

En este trabajo se abordó el tratamiento de expresiones temporales en español mediante representaciones distribuidas de las palabras, construidas a partir de la distribución de sus contextos en conjuntos de texto del orden de cientos de millones de palabras, y el entrenamiento supervisado de modelos neuronales para resolver la tarea de detección y clasificación de las expresiones.

Para llevar a cabo el trabajo se construyeron y evaluaron repertorios de representaciones para las palabras. Para la evaluación se adaptaron recursos existentes para el inglés. Se estudió además el comportamiento de las representaciones para términos del léxico de la temporalidad, detectándose que palabras dentro de un mismo grupo semántico (ej. nombre de los meses) tienden a estar próximas en distancia en el espacio de representación.

En las representaciones se observaron resultados en cuanto al ordenamiento de términos que tienen un orden preestablecido, habituales en tareas vinculadas a la temporalidad. Las representaciones de las palabras tienden a contener información del ordenamiento. Posiblemente sea consecuencia de la existencia de secuencias ordenadas de términos en el conjunto de texto a partir de donde se construyen las representaciones. Se constató que el corpus efectivamente tiene ocurrencias de secuencias ordenadas pero no se profundizó para comprobar esta conjetura. Se observó además, un comportamiento interesante, respecto a la granularidad de términos como números y ordinales. Los números de una misma granularidad tienden a estar cerca en el espacio de representación. Estas propiedades son potencialmente útiles para el tratamiento automático de expresiones temporales, siendo resultados iniciales alentadores para la viabilidad del enfoque.

Se entrenaron y evaluaron decenas de modelos neuronales, usando como entrada las representaciones previamente construidas. Se entrenaron supervisadamente para detectar y clasificar expresiones temporales. Se estudió el efecto de considerar distintos modelos y

aplicar distintas técnicas. El beneficio principal de este enfoque es que el modelo aprende a resolver la tarea utilizando únicamente datos anotados para el entrenamiento supervisado y no anotados para la construcción de las representaciones. Los modelos no incluyen ninguna información adicional del dominio ni recursos externos, se aprende únicamente de los datos.

Las variantes consideradas fueron: dimensión del espacio de las palabras, cantidad y tamaño de capas ocultas, tamaño de ventana de contexto, modelos recurrentes bidireccionales *BLSTM*s, inyección de distintos niveles de ruido en la entrada y en representaciones intermedias, distintos valores de *dropout* (máscara multiplicativa) y distintos valores de regularización *L1* y *L2*.

De las variantes consideradas se destaca la importancia de las estructurales, es decir, cantidad, tamaño y tipo de capas ocultas. Principalmente la cantidad de capas ocultas puede resultar en una mejora considerable en cuanto a la capacidad de generalización y velocidad de ajuste de la función objetivo durante el entrenamiento. Esto se muestra al comparar un modelo con una capa oculta con otro de similares características pero con dos capas ocultas. Al considerar mas de dos capas ocultas se presentan dificultades para entrenar adecuadamente el modelo.

En cuanto al contexto, la consideración de una ventana presentó mejores resultados y con un menor costo de entrenamiento que los experimentos realizados con modelos recurrentes. La consideración conjunta de una ventana de contexto en modelos recurrentes presenta ligeras mejoras pero insume un tiempo de entrenamiento mucho mayor. Es preciso realizar otros experimentos para detectar el motivo de los resultados relativamente bajos con modelos recurrentes. Queda pendiente considerar experimentos con mas unidades ocultas y pasos de recurrencia.

Respecto a las técnicas de regularización estudiadas, cabe remarcar que la mayoría de ellas presentaron alguna mejora respecto a la versión del modelo sin aplicarla. Particularmente, inyectar cierto grado de ruido en la entrada del modelo resultó en una mejora significativa, conduciendo a los mejores resultados obtenidos.

Se analizaron los resultados y se combinaron técnicas que tuvieron un efecto positivo de forma independiente. El modelo que presentó mejores resultados en la detección y clasificación de expresiones fue el modelo *feedforward* de dos capas ocultas, una de tamaño 350 (la siguiente a la capa de entrada) y la otra de 120 con 0.1 de ruido en la entrada y 0.1 de *dropout*. Este modelo presentó 73.7% de medida F en la tarea de detección y clasificación conjunta de expresiones.

En cuanto a la tarea de detección (sin clasificación), el modelo que presentó los mejores resultados, fue el modelo *feedforward* con una única capa oculta de tamaño 300 y un valor de 0.2 de ruido en la entrada. Este modelo presentó una medida F de 78.6% para el caso

estricto y 84.6% en el caso relajado, en el que se admite el corrimiento de una palabra en ambos extremos de la expresión detectada. Estos resultados mejoran al aplicar una heurística para la corrección de etiquetas inconsistentes, llegando a 79.2% de medida F para el caso estricto y 86.1% en el caso relajado.

Se estudió el comportamiento aplicar estos modelos para el inglés, donde se tiene un conjunto de entrenamiento de mayor tamaño que el considerado para los experimentos en español. El mejor modelo arrojó resultados de 78.63% de medida F en la detección y 75.24% en la detección y clasificación.

En cuanto a trabajo futuro es posible mencionar diversas lineas. Un punto central para mejorar la calidad de este trabajo es realizar una comparación adecuada con el estado del arte, evaluando sobre el mismo conjunto de evaluación y entrenando sobre el conjunto de entrenamiento completo. También corresponde validar los modelos (al menos los que dieron mejores resultados) con validación cruzada. También se propone experimentar con modelos de mayor profundidad inicializando los pesos del modelo con técnicas de pre-entrenamiento.

Respecto a las consideraciones contextuales, puede resultar interesante probar con ventanas de contexto atenuadas en los extremos, es decir, la entrada es multiplicada por un valor real entre 0 y 1 que decrece a medida que la palabra de la ventana esta más lejos de la palabra central. Otro aspecto interesante, es realizar experimentos con modelos recurrentes (no bidireccionales) considerando el contexto futuro mediante el desplazamiento de la entrada clasificada, es decir, clasificar la palabra w_{t-c} al recibir la palabra w_t como entrada. Por otro lado, sería interesante realizar experimentos con la consideración del contexto a través de modelos convolucionales.

Finalmente resta comentar el más importante de los trabajos futuros, la consideración de un modelo para la interpretación de las expresiones. Un modelo que a partir de las expresiones clasificadas e información contextual de temporalidad, como el tiempo de enunciación, realice el anclaje de las expresiones. Las propiedades vistas sobre las representaciones de las palabras alientan la posibilidad de que con la cantidad suficiente de información se pueda interpretar a las expresiones temporales exclusivamente a partir de datos anotados y no anotados, sin la inclusión de otra información.

Referencias

Adafre, S. F. y de Rijke, M. (2005). Feature engineering and post-processing for temporal expression recognition using conditional random fields. In *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*, FeatureEng '05, pages 9–16, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ahn, D., Adafre, S. F., y de Rijke, M. (2005). Towards task-based temporal extraction and recognition. In *Annotating, Extracting and Reasoning about Time and Events, 10.-15. April 2005*.

Ahn, D., van Rantwijk, J., y de Rijke, M. (2007). A cascaded machine learning approach to interpreting temporal expressions. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 420–427.

Al-Rfou, R., Perozzi, B., y Skiena, S. (2013). Polyglot: Distributed word representations for multilingual nlp. *In Proceedings Seventeenth Conference on Computational Natural Language Learning (CoNLL 2013)*.

Angeli, G., Manning, C. D., y Jurafsky, D. (2012). Parsing time: Learning to interpret time expressions. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 446–455.

Angeli, G. y Uszkoreit, J. (2013). Language-independent discriminative parsing of temporal expressions. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 83–92.

Azzinnari, A. y Martínez, A. (2016). Representación de Palabras en Espacios de Vectores. Proyecto de grado, Universidad de la República, Uruguay.

Baroni, M., Dinu, G., y Kruszewski, G. (2014). Dont count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. *In the Conference of the Association for Computational Linguistics (ACL)*.

Referencias

Becher, G., Clérin-Debart, F., y Enjalbert, P. (1998). A model for time granularity in natural language. *In Proceedings. Fifth International Workshop on Temporal Representation and Reasoning (Cat. No.98EX157)*.

Bengio, J. (2009). *Learning Deep Architectures for AI*. Foundations and Trends in Machine Learning Vol. 2, No. 1.

Bengio, Y., Ducharme, R., Vincent, P., y Janvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Bengio, Y., Simard, P., y Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*.

Bethard, S. (2013a). ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14, Atlanta, Georgia, USA. Association for Computational Linguistics.

Bethard, S. (2013b). A synchronous context free grammar for time normalization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 821–826.

Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. *In F. Fogleman-Soulie and J.Herault, editors, Neurocomputing: Algorithms, Architectures and Applications*, pages 227–236.

Bullinaria, J. A. y Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*.

Chang, A. X. y Manning, C. (2012). Sutime: A library for recognizing and normalizing time expressions. In Chair), N. C. C., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., y Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., y Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Chung, J., Gülçehre, Ç., Cho, K., y Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. Technical Report Arxiv report 1412.3555, Université de Montréal. Presented at the Deep Learning workshop at NIPS2014.

Collobert, R. y Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. *In Proceedings of ICML, pages 160–167*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., y Kuksa, P. (2011). Natural language processing (almost) from scratch. *JMLR, 12:2493–2537*.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., y Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science, 41*.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, pages 15–27.

Filannino, M. (2012). Temporal expression normalisation in natural language texts. *CoRR*, abs/1206.2010.

Filannino, M., Brown, G., y Nenadic, G. (2013). Mantime: Temporal expression identification and normalization in the tempeval-3 challenge. *CoRR*, abs/1304.7942.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., y Wolfman, G. (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems, 20(1), 116-131*.

Firth, J. R. (1957). A synopsis of linguistic theory 1930–1955. *In Studies in Linguistic Analysis, 1–32. Blackwell, Oxford*.

Graves, A. (2012). *Supervised Sequence Labelling with Recurrent Neural Networks*. Textbook, Studies in Computational Intelligence, Springer.

Graves, A. y Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks, 18(5-6):602–610*.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., y Schmidhuber, J. (2015). LSTM: A search space odyssey. *CoRR*, abs/1503.04069.

Grover, C., Tobin, R., Alex, B., y Byrne, K. (2010). Edinburgh-ltg: Tempeval-2 system description. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 333–336, Stroudsburg, PA, USA. Association for Computational Linguistics.

Han, B., Gates, D., y Levin, L. S. (2006). From language to time: A temporal expression anchorer. In *13th International Symposium on Temporal Representation and Reasoning (TIME 2006), 15-17 June 2006, Budapest, Hungary*, pages 196–203.

Han, B. y Kohlhase, M. (2003). A time calculus for natural language. In *In The 4th Workshop on Inference in Computational Semantics*.

Harris, Z. (1954). Distributional structure. *Word, 10 (23), 146–162*.

Hassan, S. y Mihalcea, R. (2009). Cross-lingual semantic relatedness using encyclopedic knowledge. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hill, F., Reichart, R., y Korhonen, A. (2014). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Preprint pubslished on arXiv. arXiv:1408.3456*.

Hinton, G. E. (1986). Learning distributed representations of concepts. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pages 1–12. Hillsdale, NJ: Erlbaum.

Hochreiter, S. y Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9(8): 1735–1780*.

Irsoy, O. y Cardie, C. (2014). Opinion mining with deep recurrent neural networks. *roceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Johnson, L., alDosari, M., Juricek, F., John, Kastner, K., Goldberg, Y., talbaumel, Yang, Y., mhr, Olson, E., y Romanov, S. (2015). theanets: v0.6.1.

Jozefowicz, R., Zaremba, W., y Sutskever, I. (2015). An empirical exploration of recurrent network architectures. *Proceedings of the 32nd International Conference on Machine Learning, Lille, France*.

Kolomiyets, O. y Moens, M.-F. (2010). Kul: Recognition and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 325–328, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lebret, R. y Lebret, R. (2013). Word emdeddings through hellinger PCA. *CoRR*, abs/1312.5542.

Lee, K., Artzi, Y., Dodge, J., y Zettlemoyer, L. (2014). *Context-dependent semantic parsing for time expressions*, volume 1, pages 1437–1447. Association for Computational Linguistics (ACL).

Levy, O. y Goldberg, Y. (2014a). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*, pages 302–308.

Levy, O. y Goldberg, Y. (2014b). Linguistic regularities in sparse and explicit word representations. *In Proceedings of the Eighteenth Conference on Computational Natural Language Learning, pages 171–180, Baltimore, Maryland*.

Levy, O. y Goldberg, Y. (2014c). Neural word embedding as implicit matrix factorization. *In Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 2177–2185*.

Levy, O., Goldberg, Y., y Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Liwicki, M., Graves, A., Fernández, S., Bunke, H., y Schmidhuber, J. (2007). Novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. *In Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR*.

Lund, K. y Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instrumentation, and Computers, Vol. 28(2), 203–208*.

Mani, I. y Wilson, D. G. (2000). Robust temporal processing of news. In *38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1-8, 2000.*

Mikolov, T., Chen, K., Corrado, G., y Dean, J. (2013a). Efficient estimation of word representations in vector space. *In Proceedings of Workshop at ICLR*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., y Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *In Proceedings of NIPS*.

Mikolov, T., tau Yih, W., y Zweig, G. (2013c). Linguistic regularities in continuous space word representations. *In Proceedings of NAACL HLT*.

Miller, G. A. y Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes, 6(1), 1-28*.

Osgood, C., Suci, G., y Tannenbaum, P. (1957). *The Measurement of Meaning*. Illini Books, IB47. University of Illinois Press.

Osgood, C. E. (1952). The nature and measurement of meaning. *Psychological Bulletin*, 49(3):197–237.

Pascanu, R., Mikolov, T., y Bengio, Y. (2012). Understanding the exploding gradient problem. *CoRR*, abs/1211.5063.

Pennington, J., Socher, R., y Manning, C. D. (2014). Glove: Global vectors for word representation. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Plaut, D. C., Nowlan, S. J., y Hinton, G. E. (1986). Experiments on learning by back-propagation. *Technical Report CMU–CS–86–126, Carnegie–Mellon University*.

Poveda, J., Surdeanu, M., y Turmo, J. (2009). An analysis of bootstrapping for the recognition of temporal expressions. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, SemiSupLearn '09, pages 49–57, Stroudsburg, PA, USA. Association for Computational Linguistics.

Puscasu, G. (2004). A framework for temporal resolution. *LREC 2004, At Lisbon, Portugal*.

Pustejovsky, J., Castaño, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., Katz, G., y Radev, D. R. (2003). Timeml: Robust specification of event and temporal expressions in text. In *New Directions in Question Answering, Papers from 2003 AAAI Spring Symposium, Stanford University, Stanford, CA, USA*, pages 28–34.

Ratinov, L. y Roth, D. (2009). Design challenges and misconceptions in named entity recognition. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning , CoNLL '09, pages 147–155, Stroudsburg, PA, USA.*

Robinson, A. J. y Fallside, F. (1987). The utility driven dynamic error propagation network. *Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department*.

Referencias

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386–408*.

Rosenblatt, F. (1963). *Principles of Neurodynamics*. Spartan, New York.

Rumelhart, D. E., Hinton, G. E., y Williams, R. J. (1986). Learning internal representations by backpropagating errors. *Nature, Vol. 323, pages 533-536*.

Sahlgren, M. (2006). *The Word-space model*. PhD thesis, University of Stockholm (Sweden).

Schuster, M. y Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 45:2673–2681*.

Strötgen, J. y Gertz, M. (2010). Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 321–324, Stroudsburg, PA, USA. Association for Computational Linguistics.

Strötgen, J. y Gertz, M. (2013). Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298.

Theano Development Team (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Tieleman, T. y Hinton, G. E. (2012). Lecture 6.5 - rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*.

UzZaman, N. y Allen, J. F. (2010). Trips and trios system for tempeval-2: Extracting temporal information from text. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 276–283, Stroudsburg, PA, USA. Association for Computational Linguistics.

UzZaman, N., Llorens, H., Allen, J. F., Derczynski, L., Verhagen, M., y Pustejovsky, J. (2012). Tempeval-3: Evaluating events, time expressions, and temporal relations. *CoRR*, abs/1206.5333.

van der Maaten, L. y Hinton, G. (2008). Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research 9(Nov):2579-2605*.

Williams, R. J. y Zipser, D. (1995). Gradient-based learning algorithms for recurrent networks and their computational complexity. *In Y. Chauvin and D. E. Rumelhart, editors, Backpropagation: Theory, Architectures and Applications. Lawrence Erlbaum Publishers*, pages 433–486.