Aprendizaje de argumentos verbales completos y su plausibilidad en oraciones a partir de corpus[1]

Hiram Calvo

Centro de Investigación en Computación-IPN,
AV. Juan de Dios Bátiz S/N esq. M.O. de Mendizábal,
Col. Industrial Vallejo, México, D. F., 07738, MEXICO.

hcalvo@cic.ipn.mx

Resumen. El aprendizaje de preferencias de argumentos de verbos usualmente se ha tratado como un problema de verbo y argumento, o a lo mucho como una relación trinaria entre sujeto, verbo y objeto. Sin embargo, la correlación simultánea de todos los argumentos en una oración no ha sido explorado a profundidad para la medida de plausibilidad de una oración debido al alto número de combinaciones potenciales de argumentos, así como a la dispersión de los datos. En este trabajo presentamos una revisión de algunos métodos comunes para aprender las preferencias de los argumentos, comenzando con el caso más simple que considera correlaciones binarias, después lo comparamos con correlaciones trinarias, y finalmente consideramos todos los argumentos. Para esto último, usamos un modelo de aprendizaje en conjunto (*ensemble learning*) mediante modelos discriminativos y generativos; mediante características de coocurrencia y características semánticas en distintos arreglos. Buscamos responder preguntas acerca del número óptimo de tópicos requeridos para los modelos de PLSI y LDA, así como el número de coocurrencias que se requiere para mejorar el desempeño. Exploramos las implicaciones de usar diversas formas de proyectar correlaciones, es decir, en un espacio de palabras, o directamente en un espacio de coocurrencia de características. Realizamos pruebas para una tarea de pseudodesambiguación aprendiendo de corpus muy grandes extraídos de Internet.

1 Introducción

Una oración puede ser vista como un verbo con múltiples argumentos. La plausibilidad de cada argumento depende no sólo del verbo, sino también de otros argumentos. Medir la plausibilidad de los argumentos del verbo se requiere en diversas tareas como el etiquetado de roles semánticos, puesto que el agrupar los argumentos del verbo medir su plausibilidad incrementa el desempeño, tal como fue mostrado por Merlo y Van Der Plus (2009) y Deschacht y Moens (2009).

El reconocimiento de metáforas requiere esta información también, puesto que podemos conocer los usos comunes de los argumentos, y un uso no común podría sugerir la presencia de una metáfora, o un error de coherencia (por ejemplo *beber la luna en*

[1] Trabajo realizado con apoyo del CONACYT-SNI, y proyecto SIP-IPN.

un vaso). La detección de malapropismos puede usar también la medida de la plausibilidad de un argumento para determinar usos incorrectos de las palabras (Bolshakov, 2005), como en *centro histérico*, en lugar de *centro histórico*, *parece un tema* tattoo, *y hemos atrapado a dos personas* auspiciosas, *está entre la* espalda *y la pared*, etc. Por otra parte, la resolución de anáforas consiste en encontrar objetos referenciados, de tal manera que se requiere, entre otras cosas, tener información a la mano de la plausibilidad de los argumentos, es decir, qué tipo de palabra satisface las restricciones de la oración, como en: *El niño juega con* eso ahí, él *come pasto, y* lo *bebí en un vaso*.

Este problema puede ser visto como recolectar una base de datos grande de marcos semánticos con categorías detalladas y ejemplos que concuerdan con estas categorías. Para este propósito, existen diversos trabajos recientes que aprovechan los recursos manualmente manufacturados como WordNet, Wikipedia, FrameNet, VerbNet o PropBank. Por ejemplo, Reisinger y Paşca (2009) anotan conceptos existentes de WordNet con atributos, y extienden las relaciones de *es-un* basándose en el modelo de análisis latente de Dirichlet (LDA) en documentos web y la Wikipedia. Yamada y otros (2009) exploran la extracción de relaciones de hipónimos de Wikipedia usando descubrimiento basado en patrones, y agrupamiento de semejanza distribucional. El problema con el enfoque de marcos semánticos para esta tarea es que los marcos semánticos son demasiado generales.

Por ejemplo, Anna Korhonen (2000) considera los verbos *volar*, *navegar* y *resbalar* como similares, y encuentra un sólo marco de subcategorización. Por otra parte, los enfoques basados en n-gramas son demasiado particulares, e incluso usando un corpus muy grande (como la web como corpus) tiene dos problemas: algunas combinaciones no están disponibles, o las cuentas tienen sentencias hacia algunas estructuras sintácticas. Por ejemplo, resolver la adjunción de frase preposicional de *extinguir fuego con agua* usando Google da *fuego con agua:* 319,000 ocurrencias; *extinguir con agua*: 32,100, resultando en la estructura *(extinguir (fuego con agua)), en lugar de (extinguir (fuego) con agua). Es por ello que requerimos de algún mecanismo para ponderar estas cuentas. Esto último ha sido llevado a cabo mediante preferencias de selección desde Resnik (1996) para preferencias de verbo a clase, y después generalizado por Agirre y Martinez (2000) para preferencias de clases de verbos a clases de sustantivos.

Trabajos más recientes incluyen a McCarthy y Carroll (2003), que desambiguan sustantivos, verbos y adjetivos usando preferencias de selección aprendidas automáticamente como distribuciones de probabilidad sobre la jerarquía de hipónimos de los sustantivos de WordNet, evaluando con Senseval-2. Sin embargo, estos trabajos mencionados tienen un problema en común, y es que consideran por separado cada argumento para un verbo.

1.1 Un argumento no es suficiente

Considere la siguiente oración:

Hay alfalfa en la granja. La vaca la come.

Quisiéramos conectar “la” con “alfalfa”, y no con “granja”. A partir de las preferencias de selección sabemos que el objeto de *comer* debería ser algo comestible, de tal

forma que sabemos que *alfalfa* es más comestible que *granja*, resolviendo este problema. A partir de marcos semánticos tenemos conocimiento similar, pero en un sentido más amplio: hay un Investor y un ingestible.

Sin embargo, esta información puede ser insuficiente en algunos casos cuando la preferencia de selección depende de otros argumentos de la frase. Por ejemplo:

La vaca come alfalfa, pero el hombre la comerá.

En este caso, no es suficiente con saber qué objeto es comestible, sino que la resolución depende de quién está comiendo. En este caso es improbable que el hombre coma alfalfa, así que la oración podría referirse al hecho de que él comerá a la vaca. Esto mismo ocurre con otros argumentos para verbos. Por ejemplo, algunos de los argumentos periféricos de FrameNet para el marco de *ingestión* son *instrumento* y *lugar*. Sin embargo, existen algunas cosas que se comen con un instrumento en particular, por ejemplo, la sopa se come con una cuchara, mientras que el arroz se come con un tenedor o con palillos, dependiendo de quién come, o el lugar donde se come. La extracción de argumentos plausibles permite construir un diccionario que funja como base de datos para este tipo de información, que puede ser vista a su vez como sentido común, puesto que es posible aprender qué tipo de actividades son desarrolladas por grupos o entidades automáticamente a partir de grandes bloques de texto.

El objetivo de nuestro trabajo es construir dicha base de datos. Para este propósito necesitamos obtener información relacionada con las preferencias de selección y la extracción de marcos semánticos.

En las siguientes secciones presentaremos el trabajo relacionado, organizado en diversos enfoques (sección 2), después presentaremos una propuesta basada en el modelo de espacio vectorial (sección 3), después una propuesta basada en modelos de lenguaje (sección 4), y finalmente presentamos nuestros aportes principales (secciones 5 y 6), que consisten en aplicar indexado probabilístico semántico latente (PLSI) para manejar tres variables correlacionadas (sección 5), y finalmente el manejo de la coocurrencia compleja mediante máquinas de soporte vectorial (SVM) a partir de características provistas por PLSI y coocurrencia (sección 6). En cada sección presentaremos diversos experimentos para mostrar cómo diferentes parámetros afectan el comportamiento del modelo, así como para comparar diversos enfoques.

1.2 Enfoques para aprender preferencias de argumentos para los verbos

El problema de aprender la plausibilidad de los argumentos de un verbo puede ser estudiado desde diversos puntos de vista. Desde el punto de vista del tipo de información extraída, podemos encontrar trabajos relacionados para preferencias de selección y extracción de marcos semánticos. Desde el punto de vista de las preferencias de selección, la tarea se enfoca en obtener automáticamente clases de argumentos para un verbo y una construcción sintáctica dados. Desde el enfoque de marcos semánticos, los argumentos se agrupan por el rol semántico que tienen, sin importar la construcción sintáctica que tengan. Este último enfoque enfatiza la distinción entre argumentos indispensables (núcleo) y periféricos. Por otra parte, podemos considerar el punto de vista de cómo esta información se representa: la tarea puede ser vista como un caso de modelado

estadístico del lenguaje, donde, dado un contexto (verbo y otros argumentos), el argumento faltante debe ser inferido con una alta probabilidad; o puede ser observado como una tarea de modelo de espacio de palabras frecuentemente visto en sistemas de recuperación de información. En las siguientes secciones presentamos trabajos relacionados a esta tarea desde estos distintos puntos de vista.

1.2.1 Preferencias de selección

La adquisición de preferencias de selección puede verse como uno de los primeros intentos para encontrar automáticamente la plausibilidad de los argumentos. Los intentos tempranos trataban con pares simples de verbo y argumento. Puesto que el recurso de aprendizaje es vasto y disperso, todos estos trabajos utilizan un mecanismo de generalización, o suavizado, para extender la cobertura. Resnik (1996) utiliza WordNet para generalizar el argumento de tipo objeto. Agirre y Martinez (2001) usan un modelo de clase a clase, de tal forma que tanto el verbo como el argumento objeto se generalizan al pertenecer a una clase usando WordNet. McCarthy y Carroll (2006) obtienen preferencias de selección como distribuciones probabilísticas aparte del argumento objeto. Padó y Lapata (2007) combinan información semántica y sintáctica estimando su modelo usando corpus con anotación de roles semánticos (por ejemplo FrameNet, PropBank), y después aplicando suavizado basado en clases mediante WordNet.

1.2.2 Marcos de subcategorización

Los siguientes trabajos tratan el problema de la extracción de la plausibilidad de argumentos de forma semisupervisada desde el enfoque de la extracción de marcos de subcategorización. Salgeiro *et al*. obtienen estructuras de argumentos de verbos. Generalizan sustantivos usando un reconocedor de entidades nombradas (IdentiFinder) y después utilizan el entorno del canal ruidoso para predecir argumentos. Ejemplos del tipo de información con la que trabajan son: *organización* compró *organización* de *organización*. *Cosa* compró las acciones en *fecha*, y a veces sin generalización, *La cafetería compró platos extras.*

Otro trabajo semisupervisado es el de Kawahara y Kurohashi (2001). Ellos generalizan utilizando un diccionario de ideas afines manualmente creado. Para encontrar los marcos de casos, usan junto con el verbo el argumento más cercano, proveyendo de desambiguación del sentido del verbo para casos similares al ejemplo que nos motivó, presentado en la sección 1.

A continuación presentaremos otros puntos de vista que tratan con la representación de la información de los argumentos del verbo.

1.2.3 El modelo de espacio de palabras, o modelo de espacio vectorial

Tradicionalmente según los modelos de recuperación de información, las palabras pueden representarse como documentos, y los contextos semánticos como características,

de tal forma que es posible construir una matriz de coocurrencia, o un espacio de palabras, donde cada intersección de palabra y contexto muestra el conteo de la frecuencia de aparición. Este enfoque ha sido usado recientemente con relaciones sintácticas (Padó y Lapata, 2007). Una cuestión importante dentro de este enfoque es la medida de semejanza elegida para comparar palabras (documentos) dadas sus características. Las medidas de semejanza comunes van desde medidas simples como la medida euclidiana, la medida coseno, y el coeficiente de Jaccard (Lee, 1999), hasta medidas como la medida de Hindle y la medida de Lin.

1.2.4 Modelo del lenguaje

Podemos ver la tarea de encontrar la plausibilidad de cierto argumento para un conjunto de oraciones como estimar una palabra dado un contexto específico. Particularmente, para este trabajo podemos considerar el contexto como las relaciones gramaticales para un verbo en particular:

$$P(w,c) = P(c) \cdot P(c|w)$$

que puede ser estimada de muchas formas. Particularmente, usando un modelo oculto de Markov, o utilizando variables latentes para el suavizado, como ya vimos con los modelos probabilísticos de indexado semántico latente (PLSI) (Hoffmann, 1999):

$$P(w,c) = \sum_{z_i} P(z) \cdot P(w|z) \cdot P(c|z)$$

La probabilidad condicional puede ser calculada a partir de conteos de frecuencia de n-gramas.

En las siguientes secciones presentaremos una propuesta simple dentro del enfoque del modelo de espacio de palabras (sección 2); posteriormente presentaremos dos algoritmos dentro del enfoque de modelo del lenguaje (sección 3).

2 Un modelo de espacio de palabras

Comenzaremos con un modelo simple para explorar las posibilidades de los últimos dos enfoques. En esta sección proponemos un modelo basado en el modelo de espacio de palabras.

Para el modelo de espacio de palabras, podemos construir una matriz donde a_2 son los renglones (documentos) y v, a_1 son características. Puesto que esta matriz es muy dispersa, usamos un diccionario de ideas afines para suavizar los valores de los argumentos. Para hacer esto, seguimos libremente el enfoque propuesto por (McCarthy *et al.*, 2004) para encontrar el sentido más frecuente, pero en este caso usamos los k vecinos más cercanos a cada argumento a_i para encontrar el predominio de una tripleta no vista dada su semejanza a todas las tripletas presentes en el corpus, midiendo la semejanza entre argumentos. En otras palabras, como en (McCarthy *et al.*, 2004, Tejada *et*

al., 2008a, 2008b) para desambiguación de los sentidos de las palabras, cada argumento semejante vota por la plausibilidad de cada tripleta.

$$Predominio(V, X_1, X_2) = \frac{\sum_{<v,a_1,a_2>\in T} sim(a_1, x_1) P_{MLE}(v, a_1, a_2)}{\sum_{<v,a_1,a_2>\in T} sim\_existe(a_1, a_2, x_1, x_2)}$$

donde T es el conjunto completo de tripletas $<$*verbo*, *argumento*$_1$, *argumento*$_2$$>$, P_{MLE} es la máxima verosimilitud de $<$*verbo*, *argumento*$_1$, *argumento*$_2$$>$, y

$$sim\_existe(a_1, a_2, x_1, x_2) = \begin{cases} 1 & \text{si } sim(a_1, x_1) \cdot sim(a_2, x_2) > 0 \\ 0 & \text{de otra forma} \end{cases}$$

Para medir la semejanza entre argumentos construimos un diccionario de ideas afines usando el método descrito por Lin (1998a) usando el analizador sintáctico Minipar (Lin, 1998b) sobre relaciones de corta distancia; es decir, previamente habíamos separado las oraciones subordinadas. Obtuvimos tripletas $<v,a_1,a_2>$ a partir de este corpus, que fueron contadas, y éstas fueron utilizadas tanto para construir el tesauro, como para ser utilizadas como fuente de coocurrencias de verbos y argumentos.

2.1 Evaluación

Comparamos estos dos modelos en una tarea de pseudodesambiguación siguiendo a Weeds y Weir (2003). Primero, obtuvimos tripletas $\langle v, a_1, a_2 \rangle$ del corpus. Después, dividimos el corpus en entrenamiento (80%) y prueba (20%). Con la primera parte entrenamos el modelo probabilístico de indexado semántico latente y creamos el modelo de espacio de palabras. Este modelo de espacio de palabras también se utilizó para obtener la medida de semejanza para cada par de argumentos. De esta forma podremos calcular la plausibilidad de $\langle v, a_1, a_2 \rangle$. Para la evaluación creamos 4-tuplas artificialmente: $\langle v, a_1, a_2, a_2' \rangle$, formadas al tomar todas las tripletas $\langle v, a_1, a_2 \rangle$ del corpus de prueba, y generando una tupla artificial $\langle v, a_1, a_2' \rangle$ eligiendo una a_2' aleatoria tal que $r_2' = r_2$, asegurándose de que esta nueva tripleta $\langle v, a_1, a_2 \rangle$creada aleatoriamente no estuviera presente en el corpus de entrenamiento. La tarea consiste en seleccionar la tupla correcta. Es posible que ocurran empates cuando ambas tuplas tienen la misma calificación (y ambas son distintas de cero). Comparamos los dos modelos, uno basado en modelos estadísticos del lenguaje (vea la sección 3) y el modelo de espacio de palabras. Utilizando el corpus de patentes de la colección NII de prueba para el sistema de recuperación de información NTCIR-5 (Fuji and Iwayama, 2005), analizamos 7,300 millones de palabras, y después extrajimos la cadena de relaciones de una forma dirigida, es decir, para la oración: X suma Y a Z por W, extrajimos las tripletas <suma, suj-X, obj-Y>, <suma, obj-Y, a-Z>, y <suma, a-Z, por-W>. Obtuvimos 706 millones de tripletas de la forma <*v*, *a1*, *a2*>. Consideramos sólo relaciones asimétricas encadenadas para evitar semejanzas falsas entre palabras que coocurren en la misma oración.

Siguiendo a Weeds y Weir (2003), elegimos 20 verbos, cubriendo verbos de alta frecuencia y verbos de baja frecuencia, y para cada uno extrajimos todas las tripletas $<v, a_1, a_2>$ presentes en el corpus de tripletas. Después realizamos los experimentos con el algoritmo basado en PLSI y el algoritmo basado en el modelo de espacio de palabras (WSM).

Experimentamos con diferentes números de tópicos para la variable latente *z* en PLSI, y con un número diferente de vecinos para el tesauro de Lin para expandir el modelo de espacio de palabras. Los resultados se muestran en la figura 2.

2.2 Análisis

Hemos mostrado resultados para un algoritmo basado en el enfoque del modelo de espacio de palabras para la extracción no supervisada de argumentos plausibles para un verbo, y lo comparamos con un enfoque probabilístico de indexado semántico latente (PLSI), encontrando evidencia particular para respaldar la afirmación de que es posible lograr buenos resultados con el método que vota por tripletas comunes usando un tesauro distribucional. Los resultados parecen ser consistentes con trabajos previos que usan diccionarios de ideas afines (Calvo *et al.*, 2005; Tejada *et al*., 2008a; 2008b): el añadir información incrementa la cobertura con poco sacrificio en cuanto a precisión.

No usamos ningún otro recurso después del analizador de dependencias, como reconocedores de entidades nombradas, o datos etiquetados para entrenar a un algoritmo de aprendizaje por computadora, así que a partir de esta etapa, el algoritmo es no supervisado.

Para desarrollar más este enfoque, es necesario experimentar con el límite superior del incremento de cobertura, puesto que cada vecino del diccionario de ideas afines está

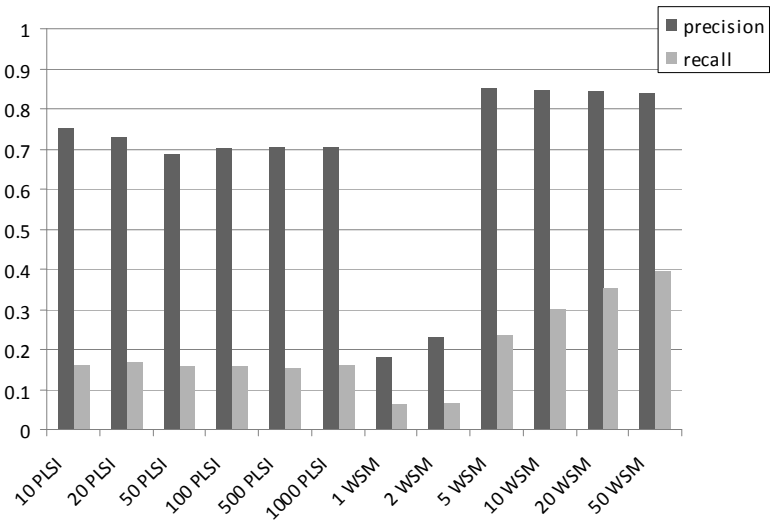

Figura 1. Resultados para (tópicos)-PLSI y (vecinos)-WSM.

añadiendo ruido. Hemos experimentado con la construcción del tesauro usando el mismo corpus; sin embargo, podrían encontrarse diferencias significativas si se usa un corpus enciclopédico para construir el diccionario, pues podría contarse con un contexto más amplio y rico.

Como trabajo futuro, también es posible experimentar con el efecto de usar otras medidas de semejanza, así como construir una tabla de semejanzas con objetos más simples: un sustantivo simple en lugar de un objeto compuesto.

En la siguiente sección exploraremos propuestas que se encuentran dentro del modelo del lenguaje.

3 Modelo del lenguaje basado en dependencias

La mayor parte del trabajo previo en modelos del lenguaje estadísticos está orientado a tareas de reconocimiento de voz (Clarkson and Rosenfeld, 1997; Rosenfeld, 2000) mediante modelos de entropía máxima. Usualmente, debido a limitaciones de espacio estos modelos se limitan a modelos secuenciales de trigramas. Diversos trabajos (Gao y Suzuki, 2003; Gao *et al.*, 2004) han mostrado que depender únicamente de n-gramas secuenciales no es siempre la mejor estrategia. Considere el ejemplo tomado de (Gao y Suzuki, 2003): *[Un bebé] [en el asiento de al lado] lloró [durante todo el vuelo]*. Un modelo de n-gramas trataría de predecir lloró a partir de *al lado*, en tanto que un modelo del lenguaje basado en dependencias (DLM por sus siglas en inglés) trataría de predecir *lloró* a partir de *bebé*.

En esta sección exploramos la creación de un DLM para obtener ocupantes de escenarios factibles, que pueden ser vistos como extraer preferencias de selección (Resnik, 1996) pero con un contexto más amplio para cada ocupante. Mostramos en la sección 3.1.1 cómo esta información adicional ayuda a obtener el mejor candidato ocupante; posteriormente en la sección 3.1.2 y 3.1.3 presentamos nuestras implementaciones de dos modelos para crear un DLM, uno basado en modelos probabilísticos de indexado semántico latente (PLSI) (sección 3.1.2) y uno basado en los k vecinos más cercanos (KNN) (sección 3.1.3). En la sección 3.2 describimos nuestros experimentos para comparar ambos algoritmos en una tarea de pseudodesambiguación. Analizaremos nuestros resultados en la sección 3.3.

3.1 Modelos para la estimación de argumentos plausibles

3.1.1 Ocupantes factibles para escenarios

Consideremos que queremos encontrar el objeto más factible de ser comido dado del verbo *comer*. Puesto que *comer* tiene diversos sentidos, el ocupante para el rol de objeto comido de *comer* podría ser comida, o podría ser un material, dependiendo de quién está comiendo. Por ejemplo, si el sujeto es ácido, entonces el objeto comido podría ser *metal*, o algún otro material (el ácido se *come* al metal).

Si se considera el problema de estimar $P(a_2|v,a_1)$ en lugar de estimar únicamente $P(a_2|v)$, donde a_1 y a_2 son argumentos, y v es un verbo, puede verse que el problema

de la dispersión de los datos se aumenta. Esto ha sido resuelto principalmente usando recursos externos como WordNet (Resnik, 1996; McCarthy and Carroll, 2006; Agirre and Martinez, 2001); recursos anotados con roles semánticos, como FrameNet, PropBank (Padó and Lapata, 2007); un reconocedor de entidades nombradas como IdentiFinder (Salgeiro *et al.*, 2006); u otros diccionarios de ideas afines manualmente creados (Kawahara and Kurohashi, 2001).

Un objetivo de esta sección es encontrar hasta qué grado la información del corpus en sí mismo puede ser utilizada para estimar $P(a_2|v,a_1)$ sin utilizar recursos adicionales. Para esto, diversas técnicas se utilizan para tratar con el problema de dispersión de los datos. Describimos dos de ellas en la siguiente sección.

3.1.2 PLSI – Modelo probabilístico de indexado semántico latente

Puesto que queremos considerar la correlación de los argumentos, usaremos la siguiente información: $P(v,r_1,n_1,r_2,n_2)$, donde *v es un verbo*, r_1 es la relación entre el verbo y n_1 (sustantivo) como sujeto, objeto, preposición o adverbio. r_2 y n_2 son análogos. Si asumimos que *n* tiene una función diferente cuando se usa con otra relación, entonces podemos considerar que *r* y *n* forman un nuevo símbolo, llamado *a*. De esta forma podemos simplificar nuestra 5-tupla a $P(v,a_1,a_2)$. Queremos saber, dado un verbo y un argumento a_1, cuál a_2 es el más plausible, es decir, queremos saber $P(a_2|v,a_1)$. Podemos escribir la probabilidad de encontrar u verbo en particular y dos de sus relaciones sintácticas como:

$$P(v,a_1,a_2) = P(v,a_1)\ P(a_2|v,a_1),$$

que puede ser estimada de distintas formas. Particularmente para este trabajo, usamos el modelo probabilístico de indexado latente semántico (Hoffmann, 1999) porque podemos explotar el concepto de variables latentes que se encargan de la dispersión de los datos.

El modelo probabilístico de indexado latente semántico (PLSI por sus siglas en inglés) fue introducido en (Hofmann, 1999), y surgió del indexado latente semántico (Deerwester et al., 1990). Este modelo intenta asociar una variable de clase no observada $z \in Z=\{z_1, ..., z_k\}$, (en nuestro caso una generalización de la correlación de la coocurrencia de v,a_1 y a_2), y dos conjuntos de observables: argumentos, y verbos+argumentos. En términos de un modelo generativo puede ser definido como sigue: se selecciona un par v,a_1 con probabilidad $P(z|v,a_1)$ y finalmente un argumento a_2 es seleccionado con probabilidad $P(a_2|z)$. Usando PLSI según (Hoffmann, 1999), es posible obtener:

$$P(\mathrm{v}, a_1, a_2) = \sum_z P(z_i)P(a_2|z_i)P(\mathrm{v}, a_1|z_i),$$

donde *z* es una variable latente que captura la correlación entre a_2 y la coocurrencia de (v,a_1) simultáneamente. Usando una variable latente para correlacionar tres variables puede conducir a un mal desempeño de PLSI, por lo que en la siguiente función exploraremos diversas formas de explotar el suavizado por variables latentes semánticas.

3.1.3 Modelo de K vecinos más cercanos (KNN-expansor)

Este modelo usa los *k* vecinos más cercanos de cada argumento para encontrar la plausibilidad de una tripleta no vista, dada su semejanza con todas las tripletas presentes en el corpus, midiendo su semejanza entre argumentos. Puesto que los votos son acumulativos, las tripletas que tienen palabras con muchas palabras semejantes tendrán más votos.

Las medidas usuales de semejanza incluyen la distancia euclidiana, coseno, y el coeficiente de Jaccard. Weeds y Weir (2003) muestran que la medida de semejanza con mejor desempeño es la medida distribucional de Lin, así que usamos esta medida para suavizar a los K vecinos más cercanos, siguiendo el procedimiento descrito por (Lin, 1998b).

3.2 Experimentos y evaluación

Para estos experimentos, usamos el mismo marco presentado en la sección 2.1. Creamos 4-tuplas artificiales $\langle v, a_1, a_2, a'_2 \rangle$, formadas tomando todas las tripletas $\langle v, a_1, a_2 \rangle$ del corpus de prueba, y generando una tripleta artificial $\langle v, a_1, a'_2 \rangle$ eligiendo una a'_2 aleatoria con $r'_2 = r_2$, asegurándose de que esta nueva tripleta aleatoria $\langle v, a_1, a'_2 \rangle$ no estuviera presente en el corpus de entrenamiento. La tarea consiste en seleccionar la tripleta correcta. Al igual que en la sección 2.1, utilizamos el corpus NTCIR-5 Patent.

3.2.1 Comparación del efecto de añadir contexto

Para este experimento, creamos un mini-corpus conjunto consistente en 1000 tripletas para cada uno de ciertos verbos elegidos del corpus de patentes: añadir, calcular, venir, hacer, comer, fijar, ir, tener, inspeccionar, aprender, gustar, leer, ver, parecer y escribir). Queremos evaluar el impacto de añadir más información para la predicción de los argumentos de los verbos, así que estimamos la plausibilidad de un argumento dado un verbo: $P(a_2|v)$; después la comparamos con el uso de información adicional de otros argumentos para ambos modelos: $P(a_2|v,a_1)$.

Para palabras completamente nuevas a veces no es posible tener un estimado, así que medimos tanto precisión como recuperación. La precisión mide cuántas adjunciones se predijeron correctamente de los ejemplos cubiertos, mientras que la recuperación mide la adjunción correcta para todo el conjunto de prueba. Nos interesa en medir la precisión y la recuperación de estos métodos, así que no implementamos ninguna técnica de retroceso.

3.3 Análisis

La operación por separado en verbos (un mini-corpus por verbo) da mejores resultados para PLSI (la precisión se encuentra arriba de 0.8), sin embargo esto parece no afectar

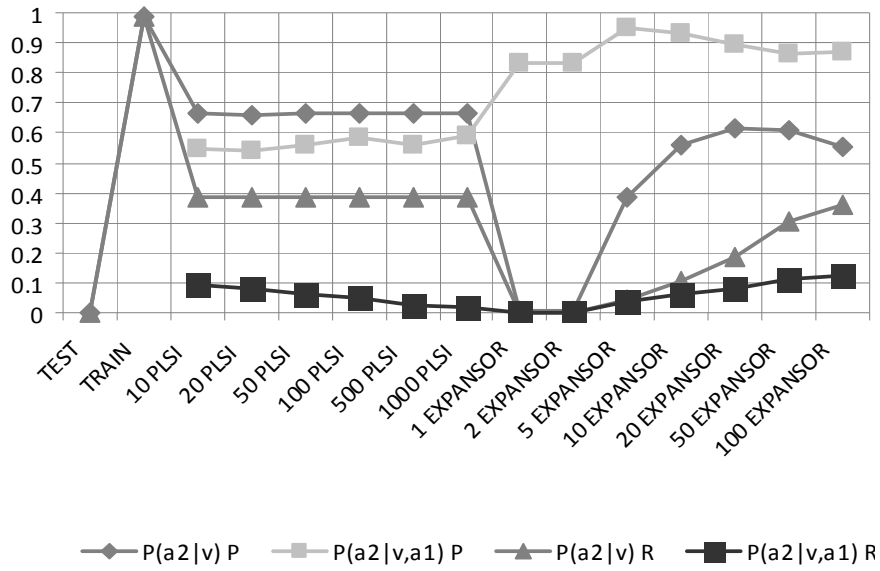

Figura 2. Efecto de añadir más contexto: predicción basa sólo en el verbo vs. predicción basada en el verbo + un argumento. KNN-Expansor es un modelo basado en *k* vecinos más cercanos

a KNN-Expansor. Para poco contexto $P(a_2|v)$, PLSI funciona mejor que KNN-Expansor. Para más contexto, $P(a_2|v,a_1)$, KNN-Expansor funciona mejor.

En general, PLSI prefiere un número pequeño de tópicos, incluso para un corpus grande (al rededor de 20 tópicos para el corpus más grande de experimentos). KNN-Expansor parece mejorar la recuperación uniformemente cuando se añaden más vecinos, perdiendo poca precisión. Expandir con pocos vecinos (1 a 5) parece no ser muy útil. Particularmente es posible ver en la Figura 2 que cuando la recuperación es muy baja, la precisión puede ser muy alta o muy baja. Esto es porque cuando se resuelven muy pocos casos, el desempeño prácticamente tiende a ser aleatoria. En general, los resultados de recuperación parecen ser bajos debido a que no utilizamos ningún método de retroceso. Si comparamos la precisión de KNN-Expansor, modelo completo (basado en más contexto), podríamos pensar que retroceder a PLSI basado en pares $P(a_2|v)$ daría mejores resultados, pero esto se ha dejado como trabajo futuro.

Evaluamos dos diferentes modelos del lenguaje basados en dependencias con una prueba de pseudodesambiguación. El modelo basado en vecinos cercanos (KNN-Expansor) se comporta mejor que PLSI cuando se incrementa la dispersión de los datos al añadir más información. Un suavizado efectivo se logra al votar usando medidas de semejanza del tesauro distribucional de Lin.

Puesto que el modelo PLSI que estamos usando tiene que manejar diversos argumentos con una sola variable latente, es posible pensar en una mejora que consiste en

interpolar diversos modelos de PLSI para manejar diversos argumentos. En la siguiente sección daremos detalles de este modelo.

4 PLSI interpolado

En esta sección proponemos un nuevo modelo llamado PLSI interpolado, que permite usar múltiples variables semánticas latentes. Este algoritmo está basado en el algoritmo descrito en la sección 3.1.2.

4.1 iPLSI – PLSI interpolado

La fórmula para PLSI previamente utilizada aglomera la asociación de información de a_2 y v, a_1 simultáneamente en una misma variable latente. Esto causa dos problemas: primero, escasez de los datos, y segundo, fija la correlación entre dos variables. De aquí que propongamos una variación para este cálculo usando interpolación basada en cada par de argumentos para una tripleta.
Una forma interpolada para estimar la probabilidad de una tripleta basada en las coocurrencias de sus diferentes pares está dada por:

$$P_E(v,a_1,a_2) \approx f_m(v,a_1) f(a_2) + f_n(v,a_2) f(a_1) + f_o(a_1,a_2) f(a_2) + f_a(v,a_1,a_2) + f_b(v,a_1,a_2) + f_c(v,a_1,a_2)$$

$$f_a(v,a_1,a_2) = \sum_a P(a_i) \cdot P(v,a_2|a) \cdot P(a_1|a)$$

$$f_b(v,a_1,a_2) = \sum_b P(b_i) \cdot P(a_1,a_2|b_i) \cdot P(v|b_i)$$

$$f_c(v,a_1,a_2) = \sum_c P(c_i) \cdot P(v,a_1|c_i) \cdot P(a_2|c_i)$$

Note que a_i (los tópicos de la variable latente) no debe ser confundida con a_1 y a_2 (los argumentos).

4.2 Experimentación

Comparemos estos dos modelos en una tarea de desambiguación, como se mostró en la sección 2.1 y 3.2. Sin embargo, para tener un rango más amplio de palabras coocurrentes, para estas evaluaciones utilizamos el corpus UKWaC (Ferraresi et al. 2008). Este corpus es un corpus grande balanceado tomado de la web, con más de 2 billones de

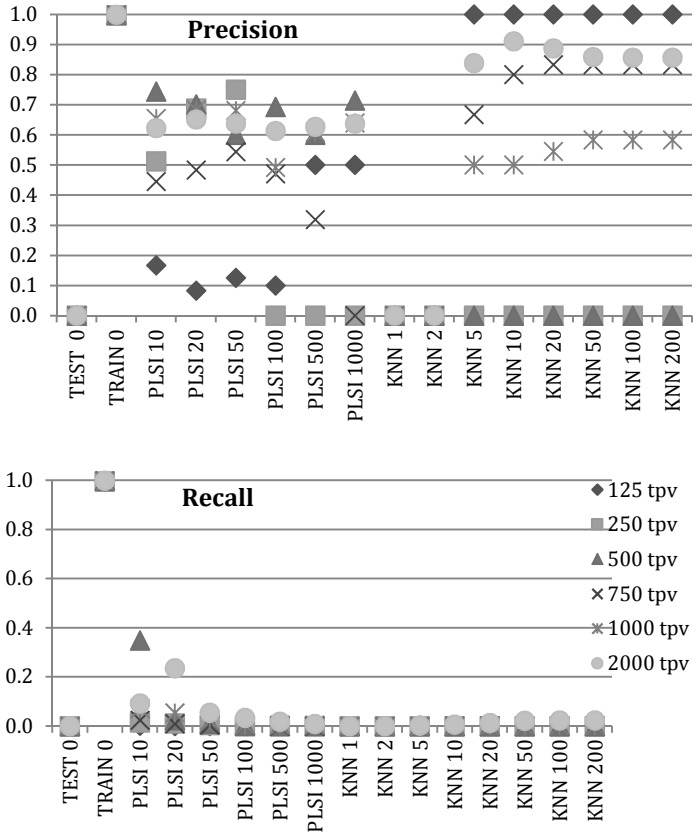

Figura 3. Promedio de precisión y cobertura para los originales PLSI y KNN-Expansor, mostrando la tasa de aprendizaje (cada serie tiene un número diferente de tripletas por verbo, tpv). No se usó umbral de frecuencia. Los números en la parte inferior muestran el número de tópicos para PLSI y el número de vecinos para KNN.

palabras[2]. Creamos dos conjuntos de palabras para los verbos: jugar, comer, sumar, calcular, fijar, leer, escribir, tener, aprender, inspeccionar, gustar, hacer, venir, ir, ver, parecer, dar, tomar, mantener, hacer, poner, enviar, decir, obtener, caminar, correr, estudiar, necesitar y devenir. Estos verbos fueron elegidos como una muestra de verbos frecuentes, y verbos poco frecuentes.

También hay verbos que pueden tomar una gran variedad de argumentos, como *tomar* (es decir, su ambigüedad es alta). Cada conjunto de palabras contiene 1000 o 2500 tripletas de verbos para cada verbo. El primer conjunto de palabras se evaluó contra 5,279 tripletas de dependencias de verbos, mientras que el segundo conjunto de palabras se evaluó con 12,677 tripletas de dependencias de verbos, correspondiendo aproximadamente al 20% del total de tripletas en cada conjunto de palabras.

[2] Una herramienta para hacer consultas de concordancias a este corpus puede encontrarse en http://sketchengine.co.uk

4.2.1 Resultados del algoritmo original con el nuevo corpus

En esta sección presentamos nuestros resultados para el nuevo corpus presentado en la sección 4.2. Las pruebas fueron llevadas a cabo con un 7 tópicos para PLSI y un valor de 100 vecinos para KNN-Expansor. Vimos en la sección 3.2 que para estimar la probabilidad de un argumento a_2, $P(a_2|v,a_1)$ funciona mejor que $P(a_2|v)$. Los experimentos realizados con este nuevo corpus confirman esto para distintos tamaños de los conjuntos de palabras. En la mayoría de los casos KNN-Expansor se comporta mejor que el PLSI original en precisión y recuperación (la mejor de las variaciones de KNN-Expansor es mejor que la mejor de las variaciones de PLSI). Contrario a KNN-Expansor, el desempeño de PLSI se incrementa en tanto que se incrementa el tamaño del conjunto de palabras, probablemente debido a que existe mayor confusión al usar el mismo número de tópicos. Esto puede ver también en la Figura 3: la recuperación mejora ligeramente para conjuntos de data mayores, y con más tópicos.

4.2.2 Medición de la tasa de aprendizaje

Este experimento consistió en incrementar gradualmente el número de tripletas de 125 a 2000 tripletas de dependencias por verbo (tpv) para examinar los efectos de usar corpus más pequeños. Los resultados se muestran en la Figura 3. En esta figura KNN-Expansor sobrepasa a PLSI cuando se añade más datos. La precisión de KNN es más alta también en general en todos los experimentos. Los mejores resultados para PLSI se obtuvieron con 7 tópicos, mientras que para KNN los mejores resultados se obtuvieron con 200 vecinos.

4.2.3 Resultados sin pre-filtrado

Los resultados anteriores usaban un umbral de pre-filtrado de 4, esto quiere decir que tripletas con menos de 4 ocurrencias fueron desechadas. Cuando quitamos este filtro, los resultados para KNN caen dramáticamente. PLSI es capaz de mantener un buen desempeño con 20 tópicos. Esto sugiere que PLSI es capaz de suavizar mejor ocurrencias simples para ciertas tripletas. KNN es mejor al trabajar con tripletas que ocurren frecuentemente. Requerimos un método que puede manejar ocurrencias de palabras no frecuentes, puesto que el pre-filtrado implica cierta pérdida de información que podría ser útil posteriormente. Por ejemplo, imagine que *tezgüino* se menciona sólo una vez en el conjunto de entrenamiento. Consideramos que es importante poder aprender información de entidades mencionadas escasamente. La siguiente sección presenta resultados con respecto a la mejora de PLSI para manejar elementos no filtrados.

4.3 Resultados de iPLSI

Como vimos en la sección 4.1, probamos diferentes modelos para combinar las variables semánticas latentes. El mejor modelo que obtuvimos combina las medidas de

(v,a_1), (a_1,a_2) y (v,a_2), respectivamente, dando una precisión de 0.83 y una recuperación de 0.83.

También realizamos pruebas con n-gramas puros (sin utilizar tripletas de dependencias, como en todas las pruebas anteriores). Veremos que los mismos componentes también dan la mejor solución.

4.4 Prueba con n-gramas

Realizamos esta prueba para corroborar que los tres componentes están contribuyendo a la interpolación, así como para evitar la tendencia que el analizador sintáctico pudiera estar provocando.

La prueba de n-gramas se realizó seleccionando trigramas de bigramas del corpus UKWaC de una forma parecida a la de los experimentos previos. Sin embargo, en este caso no usamos relaciones de dependencias, sino ventanas deslizantes de hexagramas distribuidas en trigramas como un intento de imitar la forma en la que palabras de función (como preposiciones o determinantes) afectan a las tripletas en el modelo de dependencias. Los trigramas fueron extraídos para n-gramas relacionados con los mismos verbos descritos en la sección 4.2.

La tarea consistió, como con la tarea de las tripletas de dependencias, en elegir una entre dos opciones del par 1. Usamos 80% de los trigramas como una base para entrenamiento, y el 20% para la prueba. Las pruebas se realizaron con 500 tripletas por verbo hasta 5,000 tripletas por 2000, probando todas las combinaciones posibles de elementos de (v, a_1, a_2). Al igual que en el caso anterior, combinar las medidas de (v,a_1), (a_1,a_2) y (v,a_2) tuvo el mejor desempeño (precisión de 0.77 y 0.77 de recuperación para 500 tripletas por verbo).

4.5 Análisis

Vimos que el algoritmo KNN-Expansor se desempeña mejor que el PLSI de una sola variable latente, y estudiamos la tasa de aprendizaje de ambos algoritmos, mostrando que KNN incrementa la recuperación cuando se le añaden más datos, sin perder mucha precisión; sin embargo, KNN-Expansor requiere fuertemente una fase de pre-filtrado que eventualmente conduce a una pérdida importante de palabras que ocurren escasamente.

Estas palabras son importantes para nuestros propósitos, pues quitarlas nos quita la posibilidad de generalizar palabras raras para medir su plausibilidad. El algoritmo propuesto de PLSI interpolado (iPLSI) soluciona este problema, dando mejores resultados que el PLSI de una sola variable. Encontramos que es posible seleccionar el hexagrama más factible de dos, con un 77% de recuperación para n-gramas puros agrupados como trigramas de bigramas, y hasta un 83% de recuperación para trigramas de dependencias.

Las pruebas conducidas muestran que es posible seleccionar el candidato correcto para una tripleta que puede ser vista como parte de una oración. Esto permite calcular el argumento más plausible en una oración, usando un contexto más amplio dado por un verbo y otro argumento.

iPLSI ha funcionado mejor que el modelo previo de KNN, pero aún quedan aspectos para mejorar. Particularmente, estamos estimando la coocurrencia de dos argumentos simultáneamente. Para determinar si usar más argumentos es mejor para la predicción de argumentos, proponemos un modelo que nos permite hacer esto en la siguiente sección, y después lo comparamos con los métodos previos.

5 La necesidad de medir todas las coocurrencias

Hemos visto previamente que considerar simultáneamente tres argumentos da mejor precisión que considerar únicamente dos, con cierta pérdida de recuperación. Kawahara y Kurohashi (2006) realizan desambiguación de verbos para aprender preferencias diferenciando el verbo principal con el argumento más cercano. Por ejemplo *jugar una broma* y *jugar un juego* tendrán distintas preferencias de sus otros argumentos; sin embargo, en algunos casos esto no es suficiente, como puede verse en el siguiente ejemplo, donde el verbo tiene diferentes significados dependiendo de un argumento lejano:

Poner una escena para los amigos en el teatro (montar, actuar) y
Poner una escena para los amigos en la TV (reproducir)

Trabajos recientes han propuesto un enfoque discriminativo para aprender las preferencias de selección, comenzando con Bergsma *et al.* (2008). Ritter *et al.* (2010) y Ó Séaghdha (2010) proponen LinkLDA (Latent Dirichlet Allocation), un modelo con variables de tópicos ocultas obtenidas de la misma distribución para modelar combinaciones de <sujeto, verbo, objeto> tales como <hombre, come, ramen> y <vaca, come, pasto>.

Sin embargo, estos trabajos consideran a lo más relaciones trinarias. Motivados por el problema de considerar tantos argumentos como sea posible para agrupar las preferencias de los verbos, proponemos aquí un modelo general para aprender todas las preferencias correlacionadas en una oración, permitiéndonos medir la plausibilidad de su ocurrencia. Adicionalmente, este modelo nos permite usar tanto recursos estadísticos como recursos manuales como diccionarios o WordNet para mejorar la predicción. En particular, mostraremos un ejemplo del uso de PLSI, información mutua y WordNet para medir la plausibilidad.

5.1 Método

Primeramente construimos el recurso para contar las coocurrencias. Hacemos esto, como en los casos anteriores, analizando sintácticamente el corpus UKWaC con MINIPAR (Lin, 1988) para obtener una representación lematizada de dependencias. La oración *Poner una escena para amigos en el teatro* se convierte en

```
Poner obj:escena para:amigo en:teatro.
```

Después pre-calculamos las estadísticas de información mutua entre todos los pares de palabras, por ejemplo: (poner, obj:escena), (poner, para:amigo), (poner, en:teatro),

(obj:escena, para:amigo), (obj:escena, en:teatro), (para:amigo, en:teatro). Después procedemos a calcular la representación en tópicos para cada palabra usando PLSI.

5.2 Ensamblaje de las características para entrenamiento y prueba

Una vez que se construyen los recursos de PLSI y PMI, se analizan las oraciones de entrenamiento y prueba con MINIPAR, pero sólo se utiliza el primer nivel de análisis superficial. Asignamos las características a posiciones en un vector. Cada argumento tiene una posición fija, por ejemplo, el sujeto siempre irá en la primera posición, el objeto en la posición 75, los argumentos comenzando cono *en* en la posición 150, etc. De esta manera, las correlaciones pueden ser capturadas usando aprendizaje automático. En particular, usaremos una máquina de soporte vectorial (SVM). Hemos elegido un núcleo polinomial de segundo grado, de tal forma que pueda capturar las combinaciones de características. Cada una de las características de los argumentos se descompone en diversas subcaracterísticas. Estas subcaracterísticas consisten en la proyección de cada palabra en el espacio de tópicos de PLSI, la información puntual mutua (PMI) entre la palabra objetivo y la palabra característica, y la proyección de la palabra característica en el espacio de WordNet. La información puntual mutua se calcula como sigue:

$$PMI(t_1, t_2) = \frac{\log P(t_1, t_2)}{P(t_1, t_2)}$$

5.3 Experimentos

Como en los experimentos anteriores, realizaremos una tarea de pseudodesambiguación. Esta tarea consiste en cambiar una palabra objetivo (en este caso, el objeto directo) y después el sistema identificará la oración más plausible considerando el verbo y todos sus argumentos. Por ejemplo, para las oraciones 1) como arroz con palillos en la cafetería y 2) como bolsa con palillos en la cafetería, el sistema debería ser capaz de identificar la primera como la oración más plausible. Este experimento es similar a los previos mostrados en las secciones 2.1, 3.2 y 4.2, pero en este caso estamos considerando frases completas en lugar de sólo cuádruplas. Obtuvimos al azar 50 oraciones del corpus WSJ para los verbos: jugar, comer, sumar, calcular, fijar, leer, escribir, tener, aprender, inspeccionar, gustar, hacer, venir, ir, ver, parecer, dar, tomar, mantener, hacer, poner, enviar, decir, obtener, caminar, correr, estudiar, necesitar y devenir. Estos verbos fueron elegidos como una muestra de verbos altamente frecuentes, así como de verbos poco frecuentes. También son verbos que pueden tener una gran cantidad de argumentos, como *tomar*, es decir, su ambigüedad es alta. Para el entrenamiento, creamos conjuntos de palabras para los mismos verbos. Cada conjunto de entrenamiento contiene 125, 250 o 500 tripletas de dependencias para cada verbo. Cambiar el tamaño del entrenamiento nos permite contestar a la pregunta acerca de ¿qué tanta información se requiere por cada verbo para poder aprender algo significativo?

Los conjuntos de palabras se utilizaron tanto para entrenar al modelo PLSI como para crear la base de datos PMI. Después los mismos conjuntos de palabras se utilizaron para entrenar a la máquina de soporte vectorial (SVM). Cada oración fue tratada como una línea, tal como se describe en la sección 5.2, con cada característica expandida en subcaracterísticas de PLSI (tópicos). Generamos dos ejemplos falsos aleatoriamente por cada buen ejemplo, para que la SVM tenga ejemplos de cosas correctas, como de cosas incorrectas.

5.4 Adición de información manualmente obtenida

Como se describió en la sección 5.2, añadimos información manualmente obtenida a las tablas de entrenamiento y prueba. Esta información consiste en la distancia a los 38 conceptos superiores de WordNet, según fueron propuestos por Miller: tierra, objeto, ser, humano, animal, flora, artefacto, instrumento, dispositivo, producto, escritura, construcción, trabajador, creación, comida, bebida, locación, símbolo, sustancia, dinero, ropa, sentimiento, cambio de estado, movimiento, efecto, fenómeno, actividad, acto, estado, abstracción, atributo, relación, cognición, unidad, relación, tiempo y fluido.

Los resultados de los experimentos aparecen en la siguiente tabla.

Conjunto 125							
PMI	PLSI	WN	Aprendizaje	Cobertura	Precisión	Recup.	F
0	0	1	68.36%	89.44%	54.88%	49.09%	51.82%
0	1	0	89.59%	82.61%	66.96%	55.23%	60.53%
0	1	1	92.60%	**96.09%**	63.23%	60.76%	61.97%
1	0	0	93.63%	46.62%	**70.98%**	33.10%	45.15%
1	0	1	94.55%	94.88%	65.85%	**62.48%**	**64.12%**
1	1	0	97.14%	83.03%	66.09%	54.85%	59.95%
1	1	1	**98.01%**	**96.09%**	65.26%	62.71%	63.96%
Conjunto 250							
0	0	1	67.85%	89.49%	53.87%	48.21%	50.88%
0	1	0	88.01%	87.02%	69.44%	60.43%	64.62%
0	1	1	90.82%	**96.28%**	68.22%	**65.69%**	**66.93%**
1	0	0	93.24%	55.18%	**70.34%**	38.81%	50.02%
1	0	1	93.78%	95.39%	64.86%	61.87%	63.33%
1	1	0	96.88%	87.12%	68.99%	60.11%	64.24%
1	1	1	**97.28%**	**96.28%**	66.10%	64.64%	65.36%
Conjunto 500							
0	0	1	91.09%	89.49%	46.75%	41.84%	44.16%
0	1	0	86.75%	91.58%	68.32%	62.57%	65.32%
0	1	1	93.46%	96.79%	54.37%	52.63%	53.49%

1	0	0	92.95%	64.62%	65.11%	42.07%	51.11%
1	0	1	93.46%	95.72%	63.18%	60.48%	61.80%
1	1	0	96.65%	91.72%	**68.77%**	63.08%	**65.80%**
1	1	1	**96.68%**	**97.69%**	65.51%	**63.41%**	64.44%
Promedio							
0	0	1	91.09%	89.47%	51.83%	46.38%	48.96%
0	1	0	86.75%	87.07%	68.24%	59.41%	63.49%
0	1	1	93.46%	96.39%	61.94%	59.69%	60.80%
1	0	0	92.95%	55.47%	**68.81%**	37.99%	48.76%
1	0	1	93.46%	95.33%	64.63%	61.61%	63.08%
1	1	0	96.65%	87.29%	67.95%	59.35%	63.33%
1	1	1	**96.68%**	**96.69%**	65.62%	**63.59%**	**64.59%**

A partir de estos resultados, es posible ver que en la mayoría de los casos, combinar las tres fuentes de información mejora la tasa de aprendizaje, aunque separadamente, PMI provee la tasa de aprendizaje más alta. La cobertura siempre es mejor cuando se combinan los tres recursos, sin embargo, la precisión es mejor usando sólo PMI para pequeñas cantidades de datos de entrenamiento, en tanto que PLSI da mejor soporte cuando se añade más información. La recuperación es mayor para los casos que involucran la ayuda de información de WordNet. En promedio, a excepción de la precisión, los mejores valores se obtienen cuando se combinan los tres recursos.

6 Conclusiones y trabajo futuro

A pesar de la poca cantidad de datos de entrenamiento, hemos sido capaces de obtener tazas de predicción por encima de una línea base trivial de selección aleatoria entre dos opciones. Con estos experimentos fue posible determinar el impacto de usar diversos recursos, y además de medir el beneficio de usar un modelo en conjunto para aprendizaje con máquinas de soporte vectorial, en comparación con un simple modelo probabilístico de indexado semántico latente. Encontramos que al considerar todas las co-ocurrencias de los argumentos en una oración incrementa la recuperación en un 10%. También observamos que, como se esperaba, añadir más información incrementa la cobertura; sin embargo, la recuperación se incrementa en mayor medida usando máquinas de soporte vectorial sobre los modelos probabilísticos de indexado semántico, que usando éstos últimos únicamente.

Usar SVM incrementa la cobertura, la precisión y la recuperación, incluso cuando se entrena con la misma información disponible para PLSI. Esto sugiere que generar ejemplos negativos aleatoriamente, y aplicar aprendizaje automático a esta muestra, puede mejorar el desempeño de las tareas que utilizan modelos basados en tópicos.

Hemos propuesto un modelo que integra información estadística (PLSI y PMI) con recursos manualmente producidos como WordNet, y hemos probado que el desempeño se incrementa de esta manera, aunque el incremento no fue tan significativo como esperábamos. La mayoría de las características que contribuyen al desempeño vienen de PLSI. Sin embargo, el aprendizaje automático sobre PLSI tiene la ventaja de poder

capturar la correlación entre todos los argumentos, en oposición al modelo simple de PLSI.

Los trabajos futuros derivados de éste pueden considerar explorar un modelo matemático de tres variables basado en PLSI en lugar de una interpolación por pares, así como otras variaciones de iPLSI como uno basado en dos etapas, que consistiría en relacionar dos variables semánticas latentes con una variable latente en una segunda etapa.

Puesto que la prueba que realizamos produce alternativas aleatorias, nuestro sistema podría seleccionar candidatos más probables que el actual, por ejemplo, si en el texto existiera "vaca come heno en el patio", la alternativa generada automáticamente dijera "vaca come pasto en el patio", y el sistema seleccionara la segunda como más probable, sería considerada como un error, aunque podemos ver que no es así. Aunque se espera que el efecto de esto sea despreciable, debería ser considerado en futuros análisis.

Como trabajo futuro, planeamos evaluar con conjuntos de palabras más grandes, así como evaluar el desempeño de nuestro modelo en otras tareas como resolución de anáfora o detección de coherencia de oraciones.

Referencias

1. Agirre, E. and D. Martinez. 2001. Learning class-to-class selectional preferences, *Workshop on Computational Natural Language Learning*, ACL.
2. Baroni, M. and A. Lenci. 2009. One distributional memory, many semantic spaces. *Proceedings of the EACL 2009 Geometrical Models for Natural Language Semantics (GEMS) Workshop*, East Stroudsburg PA: ACL, 1–8.
3. Bergsma, S., D. Lin and R. Goebel, 2008. Discriminative Learning of Selectional Preference for Unlabeled Text. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 59–68
4. Bolshakov, I. A. 2005. An Experiment in Detection and Correction of Malapropisms through the Web, LNCS 3406, pp. 803-815.
5. Bolshakov, I.A., S. N. Galicia-Haro, A. Gelbukh. Detection and Correction of Malapropisms in Spanish by means of Internet Search. TSD-2005, Springer LNAI 3658: 115–122, 2005.
6. Budanitsky, E., and H. Graeme. Semantic distance in WorldNet: An experimental, application-oriented evaluation of five measures, NAACL Workshop on WordNet and other lexical resources, 2001.
7. Calvo, H., A. Gelbukh, and A. Kilgarriff. Automatic Thesaurus vs. WordNet: A Comparison of Backoff Techniques for Unsupervised PP Attachment, Springer LNCS 3406:177–188, 2005.
8. Calvo, H., K. Inui and Y. Matsumoto. 2009. Interpolated PLSI for Learning Plausible Verb Arguments, In P*roceedings of the 23rd Pacific Asia Conference on Language, Information and Computation*, pp.622–629.
9. Calvo, H., K. Inui, Y. Matsumoto. 2009a. Learning Co-Relations of Plausible Verb Arguments with a WSM and a Distributional Thesaurus. Procs. of the 14th Iberoamerican Congress on Pattern Recognition, CIARP 2009, Springer, Verlag. To appear.

10. Calvo, H., K. Inui, Y. Matsumoto. 2009b. Dependency Language Modeling using KNN and PLSI. Procs. of the 8th Mexican International Conference on Artificial Intelligence, MICAI 2009, Springer, Verlag, to appear.
11. Clarkson, P. R. and R. Rosenfeld. *Statistical Language Modeling Using the CMU-Cambridge Toolkit.* Procs. ESCA Eurospeech, 1997.
12. Deerwester, S., S. T. Dumais, G. W. Furnas, Thomas K. L, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, pp. 391–407.
13. Deschacht, K. and M. Moens. 2009. Semi-supervised Semantic Role Labeling using the Latent Words Language Model. Procs. 2009 Conf. on Empirical Methods in Natural Language Processing, *Proceedings of the 2009 conference on empirical methods in natural language processing (EMNLP 2009)*, pp. 21–29.
14. Ferraresi, A., E. Zanchetta, M. Baroni and S. Bernardini. 2008. Introducing and evaluating ukWaC, a very large web-derived corpus of English. *Procs. of the WAC4 Workshop at LREC*. Marrakech, pp. 45–54.
15. Foley, W. A. Anthropological linguistics: An introduction. Blackwell Publishing, 1997.
16. Fuji A. and M. Iwayama (Eds.) Patent Retrieval Task (PATENT). Fifth NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, 2005.
17. Gao J., J. Y. Nie, G. Wu, and G. Cao, 2004. Dependence language model for information retrieval. Procs. of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 170–177, 2004.
18. Gelbukh, A. and G. Sidorov, 1999. On Indirect Anaphora Resolution. PACLING-99, pp. 181-190, 1999.
19. Hoffmann, T. 1999. Probabilistic Latent Semantic Analysis, *Procs. Uncertainity in Artificial Intelligence'99, UAI*, 289–296.
20. Jiang J. and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the International Conference on Research in Computational Linguistics* ROCLING X.
21. Kawahara, D. and S. Kurohashi, 2001. Japanese Case Frame Construction by Coupling the Verb and its Closest Case Component, 1st Intl. Conf. on Human Language Technology Research, ACL..
22. Korhonen, Anna, 2000. Using Semantically Motivated Estimates to Help Subcategorization Acquisition. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. Hong Kong. 216-223.
23. Lee, L., 1999. Measures of Distributional Similarity, Procs. 37th ACL.
24. Lin, D. 1998a. Automatic Retrieval and Clustering of Similar Words. Procs. 36th Annual Meeting of the ACL and 17th International Conference on Computational Linguistics.
25. Lin, D. 1998b. Dependency-based Evaluation of MINIPAR, Proc. Workshop on the Evaluation of Parsing Systems.
26. McCarthy, D. and J. Carroll. 2006. Disambiguating Nouns, Verbs, and Adjectives Using Automatically Acquired Selectional Preferences. *Computational Linguistics* 29-4:639–654.
27. McCarthy, D., R. Koeling, J. Weeds, and J. Carroll, Finding predominant senses in untagged text. Procs 42nd meeting of the ACL, 280–287, 2004.

28. Merlo, P. and L. Van Der Plas. 2009. Abstraction and Generalisation in Semantic Role Labels: PropBank, VerbNet or both? *Procs. 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pp. 288–296.
29. Ó Séaghdha, D. 2010. Latent variable models of selectional preference. *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics*, pp. 435–444.
30. Padó, S. and M. Lapata, 2007. Dependency-Based Construction of Semantic Space Models, Computational Linguistics 33-2: 161–199.
31. Padó, U. M. Crocker, and F. Keller, 2006. Modeling Semantic Role Plausibility in Human Sentence Processing, Procs. EACL.
32. Parton, K., K. R. McKeown, B. C., M. T. Diab, R. Grishman, D. Hakkani-Tür, M. Harper, H. Ji, W. Y. Ma, A. Meyers, S. Stolbach, A. Sun, G. Tur, W. Xu and S. Yaman. 2009. Who, What, When, Where, Why? Comparing Multiple Approaches to the Cross-Lingual 5W Task. 2009. Procs. 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pp. 423–431.
33. Ponzetto, P. S. and M. Strube, 2006. Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution, Procs. Human Language Technology Conference, NAACL, 192–199.
34. Reisinger, J and Marius Paşca. 2009. Latent Variable Models of Concept-Attribute Attachment. *Procs. 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pp. 620–628.
35. Resnik, P. 1996. Selectional Constraints: An Information-Theoretic Model and its Computational Realization, *Cognition*, 61:127–159.
36. Ritter, A., Mausam and Oren Etzioni. 2010. A Latent Dirichlet Allocation method for Selectional Preferences, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 424–434.
37. Rosenfeld, R., 2000. Two decades of statistical language modeling: where do we go from here?, Proceedings of the IEEE, Vol. 88, Issue 8, 2000, 1270–1278.
38. Salgueiro P., T. Alexandre, D. Marcu, and M. Volpe Nunes, 2006. Unsupervised Learning of Verb Argument Structures, Springer LNCS 3878, 2006.
39. Weeds, J. and D. Weir. 2003. A General Framework for Distributional Similarity, *Procs. conf on EMNLP*, Vol. 10:81-88.
40. Yamada I., K. Torisawa, J. Kazama, K. Kuroda, M. Murata, S. de Saeger, F. Bond and A. Sumida. 2009. Hypernym Discovery Based on Distributional Similarity and Hierarchical Structures. *Procs. 2009 Conf. on Empirical Methods in Natural Language Processing*, pp. 929–937.