Analizar

Grupo 2

Arequipa, Perú

2022

Analizar

Por
Grupo 2

Tesis presentada a la
Escuela Profesional de Ciencia de la Computación
de la
UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
como requisito
para obtener el título profesional
de
Licenciada en Ciencia de la Computación

Universidad Nacional de San Agustín
Facultad de Ingeniería de Producción y Servicios
Escuela Profesional de Ciencia de la Computación

Analizar

Tesis de graduación presentado por el bachiller Grupo 2 en el cumplimiento de los requisitos para obtener el título profesional de Licenciado en Ciencia de la Computación.

Arequipa, 22 de octubre del 2022

Aprobado por:

or

Prof. Dr. Wilmer Ramos Lovon
Interno
Universidad Nacional de San Agustin

A Dios, por todo lo que me ha dado, a todos los profesores por sus enseñanzas y algunos amigos.

Índice general

Índice de figuras

Índice de cuadros

Agradecimientos

En primer lugar deseo expresar mi agradecimiento al profesor Wilmer Ramos Lovon, por la dedicación y apoyo que ha brindado a este trabajo, por el respeto a mis ideas y por la dirección y el rigor que ha facilitado a las mismas.

Un trabajo de investigación es siempre fruto de ideas, proyectos y esfuerzos previos que corresponden a otras personas. En este caso mi más sincero agradecimiento a la Dr. Wilmer Ramos Lovon, de la Universidad de Nacional de San Agustin de Arequipa, con cuyo trabajo estaré siempre en deuda. Gracias por su amabilidad para facilitarnos su tiempo y sus ideas.

Pero un trabajo de investigación es también fruto del reconocimiento y del apoyo vital que nos ofrecen las personas que nos estiman, sin el cual no tendríamos la fuerza y energía que nos anima a crecer como personas y como profesionales.

Resumen

Abstract

Capítulo 1

Introducción

1.1. Contexto y Motivación

El desarrollo de aplicaciones con procesamiento de lenguaje natural abre líneas a varias posibles aplicaciones como chat boot, interpretación de textos automáticamente, desarrollo de aplicaciones de alto nivel entre otras. Para esto vimos la necesidad para que las aplicaciones la interpretación de un contexto puede cambiar completamente el sentido del contexto a una oración, para esto debemos reducir la ambigüedad en textos.

1.2. Definición del problema

La ambigüedad se refiere a términos que son estructuras gramaticales que pueden entenderse de diferentes maneras o abrirse a diferentes interpretaciones y, por lo tanto, crean dudas, incertidumbre o confusión según [Deilis Carrazana Galán, 2015].

1.3. Justificación

¿porque estas desarrollando esta tesis?

1.4. Objetivos

Analizar los métodos de reducción de ambigüedad semántica.

1.5. Objetivos especificos

1.6. Organización de la tesis

Una breve descripción de cada uno de los capítulos que estas desarrollando desde el CAP 2 hasta el capitulo antes del apéndice.

Capítulo 2

Marco teórico y Antecedentes

2.1. Desambiguación del sentido de las palabras (WSD)

En general, la desambiguación del sentido de las palabras es el problema de seleccionar un sentido de un conjunto de posibilidades predefinidas para una palabra dada en un texto o discurso. En los últimos años se han incrementado las investigaciones para crear métodos de WSD. A continuación, se describe la clasificación para métodos de WSD de acuerdo a los recursos que utilizan (figura 2.1).

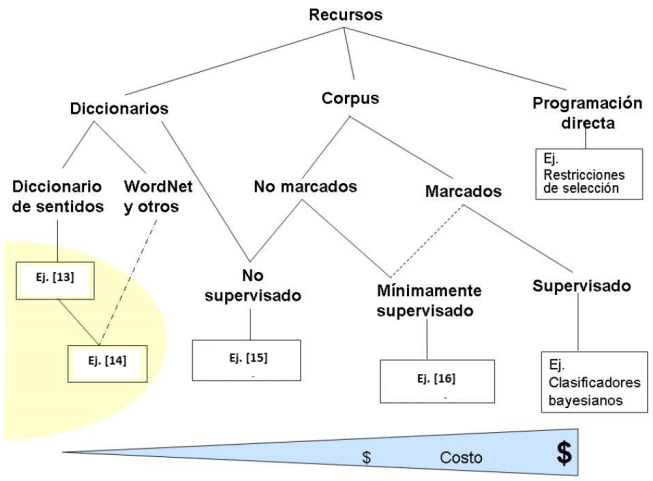

Figura 2.1: Clasificación de los métodos para WSD de acuerdo a los recursos que utilizan [Torres-Ramos, 2012].

2.2. Clasificación de sistemas en WSD

2.2.1. Métodos basados en conocimiento

En esta categoría encontramos diferentes algoritmos para la etiquetación automática de sentidos. Normalmente, el rendimiento de estos métodos basados en conocimiento, es menor en comparación con los métodos basados en corpus. Pero con la salvedad de que los métodos basados en conocimiento tienen una amplia cobertura ya que pueden aplicarse a cualquier tipo de texto en comparación con los basados en corpus que sólo se pueden aplicar a aquellas palabras de las que se dispone de corpus anotados. A continuación, vamos a enumerar diferentes técnicas utilizadas por los métodos basados en conocimiento, aplicables sobre cualquier base de conocimiento léxica que defina sentidos de palabras y relaciones entre ellas. La base de conocimiento léxica más utilizada es WordNet (Miller (1995)). Vamos a describir 4 tipos diferentes de métodos basados en conocimiento:

- Algoritmo de Lesk
- Similitud semántica
- Preferencias de selección
- Métodos Heurísticos

Algoritmo de Lesk

El algoritmo de Lesk [Lesk, 1986] es uno de los primeros algoritmos exitosos usados en la desambiguación de sentidos de palabras. Este algoritmo se basa en dos puntos principales: un algoritmo de optimización para WSD y una medida de similitud para las definiciones de los sentidos. El primer punto es acerca de desambiguar palabras considerando la coherencia global del texto, esto es, encontrar la combinación de los sentidos que maximice la relación total entre los sentidos de todas las palabras. Por ejemplo, para la oración *My father deposits his money in a bank account* y considerando a lo más tres sentidos (véase la tabla 2.1), para cada palabra, la figura 2.2 muestra la representación gráfica del algoritmo original de Lesk.

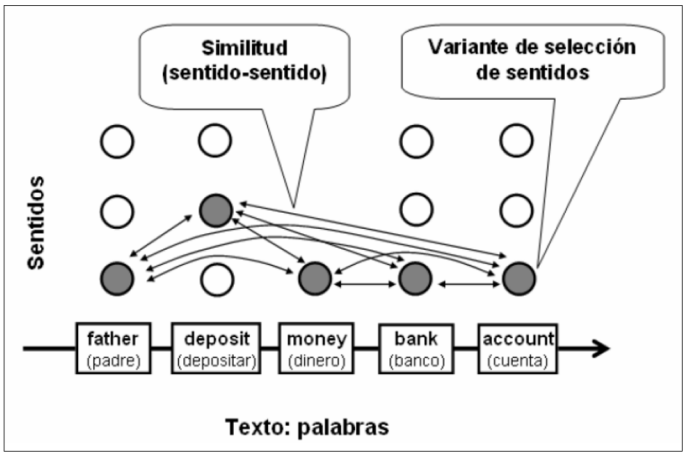

Figura 2.2: Representación gráfica del algoritmo original de Lesk [1]

Palabra	Sentidos
Father	1: a male parent (also used as a term of address to your father); "his father was born in Atlanta". 2: 'Father' is a term of address for priests in some churches (especially the Roman Catholic Church or the Orthodox Catholic Church); "'Padre' is frequently used in the military". 3: a person who holds an important or distinguished position in some organization; "the tennis fathers ruled in her favor"; "the city fathers endorsed the proposal".
Deposit	1: fix, force, or implant; "lodge a bullet in the table". 2: put into a bank account; "She deposits her paycheck every month". 3: put (something somewhere) firmly; "She posited her hand on his shoulder"; "deposit the suitcase on the bench"; "fix your eyes on this spot".
Money	1: the official currency issued by a government or national bank; "he changed his money into francs".
Bank	1: a financial institution that accepts deposits and channels the money into lending activities; "he cashed a check at the bank"; "that bank holds the mortgage on my home". 2: sloping land (especially the slope beside a body of water); "they pulled the canoe up on the bank"; "he sat on the bank of the river and watched the currents". 3: a supply or stock held in reserve for future use (especially in emergencies)
Account	1: a formal contractual relationship established to provide for regular banking or brokerage or business services; "he asked to see the executive who handled his account". 2: the act of informing by verbal report; "he heard reports that they were causing trouble"; "by all accounts they were a happy couple". 3: a record or narrative description of past events; .ª history of France"; "he gave an inaccurate account of the plot to kill the president"; "the story of exposure to lead".

Cuadro 2.1: Sentidos de las palabras (máximo tres) obtenidas de WordNet para la oración *"My father deposits his money in a bank account"*.[1]

Sentido elegido para *deposit*	Sentido elegido para *bank*	Valor de relación (traslape de palabras)
1	1	0
1	2	0
1	3	0
2	1	2
2	2	1
2	3	0
3	1	1
3	2	0
3	3	0

Cuadro 2.2: Sentidos de las palabras (máximo tres) obtenidas de WordNet para la oración *"My father deposits his money in a bank account"*.[1]

En el segundo punto, relacionado con la medida de similitud, Lesk sugiere usar el traslape entre las definiciones de los sentidos, es decir, contar el número de palabras que tienen en común. Como ejemplo, para la oración, *"My father deposits his money in the bank ac-count"* para medir la relación de las definiciones de los sentidos para la palabra *"de-posit"* y *"bank"* como Lesk lo propuso, es necesario contar las palabras en común en todas las definiciones. En este caso, comparando principalmente las tres definiciones de *"deposit"* contra las tres definiciones de "bank". La relación entre los valores se muestra en la tabla 2.2.

Este algoritmo tiene dos limitaciones, por un lado, la limitación principal de la medida de similitud propuesta por Lesk, es que las glosas del diccionario, regularmente, son muy cortas y no incluyen el vocabulario suficiente para identificar los sentidos relacionados [3]. Por otro lado, mientras más palabras tenga el texto, y más sentidos por cada palabra, mayor será el número de combinaciones de sentidos, haciéndolo prácticamente prohibitivo para una búsqueda exhaustiva que garantice encontrar el óptimo global exacto. Por ejemplo, para una oración de 16 palabras de contenido, donde cada palabra contiene siete sentidos (números cercanos a los observados en el corpus de SemCor), existen 716 posibles combinaciones a escoger, de las cuales una será seleccionada. Debido a estas dos limitaciones, diferentes modificaciones al algoritmo original han sido propuestas para mejorar los resultados en la desambiguación de sentidos de palabras, las cuales se describen en la siguiente sección.

- **Lesk simple o Lesk simplificado**

  Para reducir el espacio de búsqueda del algoritmo original de Lesk, Kilgarriff y Rosenzweig [4] propusieron una variación del algoritmo original de Lesk, conocido como algoritmo de **Lesk simplificado o Lesk simple**, donde los sentidos de las pala-bras en el texto son determinados uno a uno encontrando el mayor traslape entre los sentidos de las definiciones de cada palabra con el contexto actual, véase la figura 2.3. En lugar de buscar asignar, simultáneamente, el significado de todas las palabras en un texto dado, este enfoque determina el sentido de las palabras uno a uno, por lo que se evita la explosión combinatoria de sentidos.

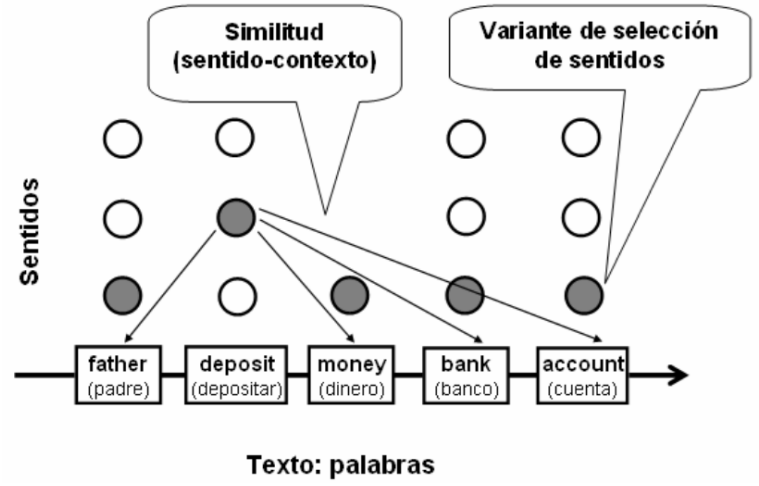

Figura 2.3: Representación gráfica del algoritmo de Lesk simplificado [1]

- **Templado simulado (Simulated Annealing)**

  El método de templado simulado es una técnica para la resolución de problemas de optimización combinatoria a gran escala. El nombre de este algoritmo es una analogía del proceso metalúrgico en el cuál, el metal se enfría y se templa. La característica de este fenómeno es que en el enfriamiento lento alcanza una composición uniforme y un estado de energía mínimo, sin embargo, cuando el proceso de enfriamiento es rápido, el metal alcanza un estado amorfo y con un estado alto de energía. En templado simulado la variable **T** corresponde a la temperatura que decrece lentamente hasta encontrar el estado mínimo. El proceso requiere una función **E**, la cual representa el estado de energía de cada configuración del sis-

tema. Es esta función la que se intenta minimizar. A grandes rasgos el algoritmo funciona de la siguiente manera: se selecciona un punto inicial y además se escoge otra configuración de manera aleatoria, se calcula para ambas con-figuraciones su valor **E**, si el nuevo valor es menor que el seleccionado como punto inicial, entonces el inicial es remplazado por la nueva configuración. Una característica esencial del templado simulado es que, existe el caso en el que la nueva configuración es mayor a la configuración obtenida anteriormente, y la nueva es seleccionada. Esta decisión es tomada de manera probabilística y permite salir de algún mínimo local. Una vez que el método mantenga la misma configuración por un determinado tiempo, dicha configuración es escogida como la solución. Cowie et al. [5], basándose en el algoritmo de Lesk, utilizó este método para desambiguación de sentidos de palabras de la siguiente forma:

1. El algoritmo define una función **E** para la combinación de sentidos de palabras en un texto dado.
2. Se calcula **E** para la configuración inicial **C**, donde **C** es el sentido más frecuente para cada palabra.
3. Para cada iteración, se escoge aleatoriamente otra configuración conocida como **C'**, y se calcula su valor de **E**. Si el valor de **E** para **C'** es menor que el de **C** entonces se elige **C'** como configuración inicial.
4. La rutina termina cuando la configuración de sentidos no ha cambiado en un tiempo determinado.

- Medida de Lesk Adaptada Lesk propuso medir la similitud entre sentidos contando el traslape de palabras. La limitación principal de esta técnica es que las glosas del diccionario, por lo general, son muy breves, de tal manera que no incluyen suficiente vocabulario para identificar los sentidos relacionados. En [6] se sugiere una adaptación del algoritmo basado en WordNet. Esta adaptación consiste en tomar en cuenta las glosas de los vecinos de la palabra a desambiguar, explotando los conceptos jerárquicos de WordNet, de tal ma-nera que las glosas de los vecinos son expandidas incluyendo a su vez las glosas de las palabras con las cuales se encuentran relacionadas mediante las diversas jerarquías que presenta WordNet.

Así mismo, sugieren una variación en la manera de asignar el puntaje a una glosa, de tal manera que si "n" palabras consecutivas son iguales en ambas glosas, estas deberán de tener mayor puntaje que aquel caso en el que sólo coincide una sola palabra en ambas glosas. Supongamos que *bark* (ladrido o corteza) es la palabra que se desea desambiguar y sus vecinos son textitdog (perro) y textittail (cola). El algoritmo original de Lesk verifica las coincidencias en las glosas de los sentidos de textitdog con las glosas de textitbark. Luego verifica las coincidencias en las glosas de textitbark y textittail. El sentido de textitbark con el máximo número de coincidencias es seleccionado. La adaptación del algoritmo de Lesk consi-dera estas mismas coincidencias y añade además las glosas de los sentidos de los conceptos que se encuentran relacionados semántica o léxicamente a textitdog, textitbark y textittail, de acuerdo a las jerarquías de WordNet.

Similitud semántica

Preferencias de selección

Métodos Heurísticos

2.2.2. Métodos basados en corpus no supervisados

El desarrollo de métodos que tratan de resolver el problema de la ambigüedad léxica ha supuesto la aparición de diferentes algoritmos que utilizan una serie de recursos diferentes. Podemos encontrar desde sistemas que utilizan técnicas de enriquecimiento de conocimiento utilizando diccionarios, tesauros o jerarquías de conceptos (los llamados basados en conocimiento), hasta sistemas que utilizan la información de textos anotados semánticamente (los llamados sistemas supervisados basados en corpus). El único inconveniente de estos sistemas es que es necesario la creación de textos, diccionarios u otras fuentes de información, de forma manual. Esto supone un gran costo en su obtención y mantenimiento, además de generar dificultades cuando se tratan de anotar textos muy extensos, de un nuevo dominio o de un lenguaje diferente. Para evitar esta dependencia se han desarrollado dos aproximaciones diferentes. La primera de ellas trata de establecer distinciones entre sentidos basándose en su distribución, determinando por tanto que, palabras que aparecen en contextos similares deben tener sentidos similares [7]y [8].

La segunda aproximación está basada en equivalencias de traducción en corpus paralelos, los cuales identifican traducciones de una palabra en un lenguaje determinado cuya traducción depende del sentido de esa palabra en el lenguaje origen. Estas traducciones dependientes del sentido de una palabra pueden ser utilizadas como una recopilación de sentidos para esa palabra en el lenguaje origen. Una de las claves de los métodos basados en distribución, es que no utilizan ningún recopilatorio de sentidos, únicamente clasifican palabras basándose en sus contextos observados en los corpus. Esta es una alternativa a los métodos que dependen de la anotación de corpus y que están restringidos a aquellas palabras que un experto ha clasificado para sus distintos sentidos. En todo caso, a pesar de que exista un repertorio de sentidos, su utilidad depende de las aplicaciones sobre las que se aplique. Las aproximaciones distribucionales no asignan sentidos a las palabras, pero sí permiten discriminar entre los sentidos de una palabra identificando clusters en contextos similares, donde cada cluster muestra que una palabra se está utilizando con un sentido determinado. Estos métodos presentan una aproximación diferente a la tarea tradicional de WSD, la cual clasifica palabras con respecto a un repertorio de sentidos existente. Los métodos basados en equivalencias de traducción se basan en el hecho de que los sentidos diferentes de una palabra en un lenguaje origen se pueden traducir en palabras diferentes en el lenguaje destino. Estas aproximaciones tienen dos propiedades. Primero, automáticamente derivan un repertorio de sentidos que hace distinciones relevantes para los problemas de traducción automática. Segundo, un corpus etiquetado basado en estas distinciones puede ser creado automáticamente y utilizado como corpus de entrenamiento para los métodos tradicionales de aprendizaje supervisado. Una de las ventajas de utilizar métodos no supervisados basados en corpus, es que no se basan en ningún diccionario, repositorio de sentidos, tesauro, etc. De forma que no están restringidos a la interpretación de sentidos que el autor del diccionario haya impuesto. Ya que, es muy habitual que diferentes diccionarios aporten una distinción de sentidos más fina o más compacta, según la finalidad para la que estén creados. Al evitar hacer uso de estos recursos, se garantiza la adaptabilidad de estos sistemas a diferentes campos o ámbitos. Otra ventaja no menos importante, es que estos métodos son independientes del lenguaje. Es decir, son fácilmente adaptables a cualquier idioma que disponga de un corpus sobre el que obtener información.

Métodos distribucionales

Este tipo de métodos identifican las palabras que suelen aparecer en contextos similares, sin necesidad de utilizar un repositorio de sentidos. En [9] por ejemplo, se realiza el proceso de desambiguación en dos pasos. El primer paso, es construir clusters que comparten características similares. El segundo paso, es etiquetar cada cluster con una definición que establezca el sentido de la palabra dentro de ese contexto. Esta es una visión completamente diferente del concepto general de WSD, donde los sentidos se suponen conocidos antes de comenzar el proceso de desambiguación. Esta nueva visión de "discriminación y etiquetación" corresponde a la forma ideal de obtener la definición de una palabra (lexicografía). Un lexicógrafo, seleccionaría diferentes contextos de una palabra determinada, a partir de un corpus extenso y representativo para el usuario final. Por ejemplo, si hablamos de un diccionario para niños, el corpus consistiría en textos escritos para niños. Y si hablamos de un diccionario sobre un dominio específico el corpus deberían ser textos de esa especialidad en particular. De esta forma el lexicógrafo, dividiría los contextos en los que aparece la palabra a estudiar en diferentes clusters, discriminando los diferentes sentidos que puede adoptar esa palabra, sin tener ninguna idea preconcebida de cuántos sentidos puede adoptar. El resultado de la discriminación es un número determinado de clusters que establecen los diferentes sentidos de la palabra, obtenidos estos a partir del corpus de entrada. A partir de aquí, se debe estudiar cada cluster y obtener una definición que actúe como una etiqueta o un sentido específico para la palabra. Esta última parte, la de asignar una definición concreta a la palabra en cada cluster es la más problemática, dado que en muchas ocasiones es difícil establecer una definición a partir de los contextos. Una posible solución sería identificar el conjunto de palabras que aparecen en un cluster y utilizarlas como una aproximación al sentido de la palabra. Por ejemplo, si tenemos la palabra "línea" y un cluster con: "teléfono", "llamada", "ocupada", "móvil". En este caso, estas palabras son indicativas del sentido asociado a este cluster. De esta forma, si los métodos no supervisados basados en corpus son desarrollados eficientemente, el resultado podría llegar a ser un proceso independiente del lenguaje que resuelve el problema de la ambigüedad sin tener que recurrir a un repositorio de sentidos. Existen dos aproximaciones distintas para los métodos

distribucionales: Discriminación basada en tipos. Estos métodos identifican conjuntos (o clusters) de palabras que pueden estar relacionadas entre sí debido a su aparición en contextos similares. Normalmente se basan en medidas de similitud entre vectores de co-ocurrencia. Discriminación basada en tokens. Estos métodos agrupan todos los contextos donde una palabra determinada aparece, basándose en la similitud de estos contextos.

Discriminación basada en tipos

En el caso de los métodos de discriminación basados en tipos, es necesario disponer de corpus extensos para poder extraer la similitud entre diferentes contextos donde aparece la palabra a desambiguar. En estos métodos la representación más utilizada se basa normalmente en la contabilización de co-ocurrencias o en medidas de asociación entre palabras. Usando esta información es posible identificar otras palabras que aparecen en contextos similares y por tanto pueden tener sentidos similares. De esta forma, se pueden extraer los distintos sentidos que puede adoptar una palabra polisémica. Por ejemplo, si seleccionamos la palabra "línea" que puede tener varios sentidos (línea telefónica, trazo, premio en el juego del bingo, etc), y ésta aparece en dos contextos distintos: contexto1 (dibujo, trazo, color, coordenada) y contexto2 (auricular, teléfono, comunicar, llamada). Podemos establecer a partir de las palabras extraídas del contexto, que en el primer caso "línea" hace referencia a un trazo en un dibujo, y en el segundo caso, hace referencia a una línea telefónica. Como ya se ha mencionado anteriormente, los métodos distribucionales basados en tipos necesitan de corpus bastante extensos. Es por ello, que la representación del espacio contextual se realizará en matrices de NxN dimensiones, donde N, es el número de palabras en el corpus. Cada celda de esta matriz contiene el número de veces que las palabras representadas en cada columna y fila co-ocurren dentro de una ventana de un tamaño especificado. Cuando no importa el orden en el que aparecen las palabras la frecuencia será la misma, pero si hablamos de bigramas, donde el orden sí importa, el valor de las celdas será distinto. Por tanto, si el orden no importa, se tendrá cuadrada y simétrica. Sin embargo, si tenemos en cuenta el orden de aparición de las palabras, tendremos una matriz rectangular y no simétrica. Para estas matrices de co-ocurrencia, las celdas pueden almacenar el número de veces que dos

	Corpus 1	Corpus 2	Total
Frecuencia de la palabra	a	b	a + b
Frecuencia de otras palabras	c - a	d - b	c + d - a - b
Total	c	d	c + d

Cuadro 2.3: Tabla 2x2 para log-likelihood ratio [12]

palabras co-ocurren, o también pueden tomar valores más complejos. Por ejemplo, las celdas de una matriz de co-ocurrencia pueden contener el valor de diferentes medidas de asociación como: log-likelihood ratio [10] o Información Mutua [11]. Estas medidas indican el grado en que dos palabras co-ocurren con respecto a las demás palabras del corpus. En el caso de la medida del log-likelihood ratio partimos de una tabla 2 x 2 como sigue a continuación Tabla 2.3.

En la Tabla 2.3 se extraen las frecuencias relativas de una palabra entre dos corpus. Se denota por c al número de palabras total del corpus1 y por d al número de palabras total del corpus2 (N en total). Los valores de a y b son denotados como valores observados (O). Por último, queda por definir los valores esperados (E), según la Fórmula 2.1

$$E_i = \frac{N_i \sum_i O_i}{\sum_i N_i} \tag{2.1}$$

Para la Tabla 2.3 $N_1 = c$ y $N_2 = d$. Por lo tanto, para la palabra que estamos tratando:

$$E_1 = \frac{c * (a+b)}{(c+d)} \qquad \text{y} \qquad E_2 = \frac{d * (a+b)}{(c+d)} \tag{2.2}$$

Los cálculos para obtener los valores esperados tienen en cuenta el tamaño de los dos corpus. Por tanto, no es necesario normalizar los valores, pudiendo aplicar a continuación la medida del log-likelihood según la Fórmula 2.3

$$-2 \ln \lambda = 2 \sum_i O_i \ln \left(\frac{O_i}{E_i} \right) \tag{2.3}$$

La Fórmula 2.3 es equivalente a calcular el log-likelihood ratio G^2 como sigue:

$$G^2 = 2 * \left(a * \ln \left(\frac{a}{E_1} \right) \right) + \left(b * \ln \frac{b}{E_2} \right) \tag{2.4}$$

Si los valores esperados y los observados son comparables, el valor de G^2 estará próximo a 0, lo que significa que la palabra ha aparecido junto a otra por casualidad, y no están relacionadas entre sí. Si se obtiene un valor mayor que 0, significa que los valores observados difieren en gran medida de los valores esperados, por lo que las palabras estarán fuertemente relacionadas entre sí. Una vez decidido el tipo de medida a utilizar para establecer la co-ocurrencia entre distintas palabras y construida la matriz de co-ocurrencia, cada palabra será representada como un vector de N-dimensiones. A partir de cada vector obtenido, se puede medir la similitud contextual entre dos palabras obteniendo el coseno entre los vectores. Para el cálculo del coseno entre dos vectores se utiliza la Fórmula 2.5

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \times |\vec{y}|} \tag{2.5}$$

Continuando con la definición de métodos distribucionales basados en tipos, encontramos distintos algoritmos que pueden ser aplicados. En esta sección vamos a tratar dos de estos algoritmos: Análisis de la Semántica Latente (LSA) [13] y Clustering by Committee (CBC) [14]. Mediante el algoritmo de LSA se representa un corpus en un espacio multidimensional, usando vectores. Cada vector representará el contexto en el cual aparece una palabra. En el caso de LSA no se hacen distinciones entre los distintos sentidos de una palabra polisémica, es decir, se formará un único vector para cada palabra, aunque ésta tenga varios sentidos diferentes. Usando la información del contexto, se podrá determinar, por ejemplo, que palabras como: coche, automóvil, auto... están relacionadas semánticamente.

Cuando hablamos de LSA, debemos pensar en la representación del conocimiento como matrices de [palabras-contextos]. Para medir el grado de similitud de una palabra con respecto a otras palabras del contexto, se utiliza la medida del coseno entre vectores. Además de poder comparar palabras y contextos, también se puede medir el grado de similitud entre oración-oración, contexto-contexto... simplemente calculando el vector resultado de la unión de cada uno de los vectores que conforman las palabras

de la oración, del contexto, etc. Mediante el algoritmo de CBC se pueden detectar clusters de palabras relacionadas con los distintos sentidos de una palabra polisémica. Por ejemplo, para la palabra "muñeca" el algoritmo de CBC podría identificar dos clusters, uno asociado con el sentido de juguete, con palabras como juego, entretenimiento, niños, cochecito, etc, y otro cluster asociado con el sentido de parte del cuerpo humano, con palabras como brazo, extremidad, articulación, etc. Por lo tanto, con el algoritmo de CBC se pueden detectar palabras sinónimas asociadas a los diferentes sentidos de una palabra. Ambos algoritmos, tanto LSA como CBC, utilizan representaciones multidimensionales de co-ocurrencia de palabras.

Discriminación basada en tokens

El objetivo de este tipo de métodos es agrupar los contextos en los que una palabra aparece bajo el mismo sentido. A continuación, se van a describir métodos que utilizan características de primer y segundo orden. Las características de primer orden ocurren directamente en un contexto que está siendo clasificado, mientras que las características de segundo orden son aquellas que ocurren junto con una de primer orden, pero no ocurren en el contexto que está siendo clasificado. En primer lugar, es necesario establecer cómo se van a representar los contextos que van a ser clasificados en clusters. Al igual que para los sistemas supervisados, los contextos contienen la palabra a desambiguar con la excepción de que esta no tiene asignado ningún sentido. La premisa de los métodos basados en tokens es que si una palabra aparece en contextos similares ésta ha de tener el mismo sentido. Uno de los primeros métodos que utilizó discriminación basada en tokens fue una adaptación del algoritmo de LSA usando características de segundo orden [9]. En este caso, la representación de la matriz de co-ocurrencia en lugar de utilizar palabras utiliza contextos completos usando co-ocurrencias de segundo orden de características léxicas. Una palabra tiene una co-ocurrencia de segundo orden con otra, cuando ambas no aparecen juntas, pero ambas sí aparecen junto a otra palabra frecuentemente. Por ejemplo, en "policía de tráfico" y "accidente de tráfico", la palabra "policía" es una co-ocurrencia de segundo orden de "accidente", porque ambas co-ocurren en primer orden con "tráfico". Otro método que utiliza esta aproximación es el de [15]. En este caso, utilizan un conjunto reducido de características de primer orden para crear

matrices que muestran la similitud entre contextos. Estas características se extraen a partir de las palabras que se encuentran alrededor de la palabra a desambiguar e incluyen etiquetas sintácticas y palabras co-ocurrentes. El problema de este tipo de métodos es la forma de evaluación de los resultados. Debido a que la discriminación no parte de un conjunto preestablecido de sentidos, no se puede evaluar la forma en que los nuevos sentidos son descubiertos, sobretodo si se está trabajando en un dominio específico.

2.2.3. Métodos basados en corpus supervisados

2.2.4. Métodos híbridos

Capítulo 3

Formalismos y/o teoría propuesta

Seria recomendable que cada uno de las etapas o puntos principales vayan acompañados de una discusión (mini conclusión) detallando y justificando la razón de su existencia.

3.1. Instalación de LaTeX

Debemos iniciar la instalación mediante los siguientes paquetes básicos, es recomendado seguir el siguiente orden en la instalación:

AFPLGhostscript Nos permite trabajar con los formatos EPS que caracterizan a LaTeX(Free).

GSview Para visualizar los PS y EPS

Acrobat Reader Para visualizar los PDF (Free).

small-miktex el compilador y los `packages` del LaTeX(Free).

WinEdt Un potente editor para LaTeX.

Estos paquetes son opcionales, pero muy útiles:

Diccionario Diccionario para poder corregir en español, aún incompleto solo en WinEdt (Free).

GNUplot Poderoso Graficador y procesador matemático, muy usado en los trabajos de investigación y tesis(Free).

Capítulo 4

Experimentación o evaluación empírica

4.1. Notación Matemática

Esta sección contiene un curso ultra rápido de como escribir fórmulas matemáticas en tus documentos. Vamos a revisar únicamente algunas construcciones sencillas y frecuentes.

$\lim_{n\to\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$

$$\forall x \in \mathbf{R}: \qquad x^2 \geq 0 \tag{4.1}$$

$$\underbrace{a + b + \cdots + z}_{26}$$

$$\iint_D g(x, y)\,\mathrm{d}x\,\mathrm{d}y$$

en lugar de

$$\int\int_D g(x, y)\mathrm{d}x\mathrm{d}y$$

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

Capítulo 5

Resultados y/o Evaluaciones

5.1. Conclusiones

Conclusiones que han podido ser cuantificadas o ampliamente deducibles (criterio lógico general) en base a TU TESIS.

5.2. Contribuciones

Con todo lo que has investigado, propuesto y/o desarrollado que haz conseguido obtener para cooperar con la solución del problema.

5.3. Trabajo futuro

Bueno ha estas alturas definitivamente te habrás dado cuenta que existe un montón de problemas que directa o indirectamente necesitan ser solucionados, es recomendable solo proponer y mostrar aquellos que en tu tesis haya una viabilidad cercana o muy relacionada, de tal forma pueda que una futura tesis(compañero) o estudios superiores le den continuidad.

Apéndice A

Formato de la plantilla

Antes de seguir debes de tener en tu computadora la clase `unsa.cls` y demás archivos relacionados. En esta clase se encuentra la información sobre el formato casi oficial de la tesis en la EPIS UNSA, así como muchos comandos e instrucciones especiales que facilitarán tu existencia mientras escribes tu tesis. He puesto todos los archivos necesarios en Internet, en `http://www.spc.org.pe/tutoriales/tesis-pregrado/` , a disposición de todo el que esté interesado. El camino más sencillo es descomprimir todos los archivos que vienen en el paquete dentro de una nueva carpeta en tu computadora donde guardes normalmente tus archivos (por ejemplo: `Mis Documentos\Tesis\`).

A.1. Datos de la tesis

Entre los archivos incluidos en el paquete encontrarás un `Tesis.tex` que puedes abrir en tú editor de texto, aunque es recomendable usar el WinEdt (en esta versión de la plantilla se uso éste software).

Cuando hayas abierto el archivo verás algunos comandos, quizá la mayoría de ellos desconocidos, pero no te preocupes demasiado por eso en este momento. Por ahora, como estoy suponiendo que quieres empezar lo más pronto posible a escribir tu tesis, no voy a analizar de manera detallada el contenido de este archivo. Voy a ir directamente sobre lo que sí debes de saber para poder comenzar. Como podrás ver, en el archivo

hay un grupo de líneas de de la forma:

```
\documentclass[11pt,openright,final]{tuuniversidad}
\title{Escribe el titulo de tu tesis}
\author{Escribe tu nombre}
\examinerone{Nombre del Presidente}{Presidente}
\examinertwo{Nombre del Secretario}{Secretario}
\examinerthree{Nombre del Integrante}{Integrante}
\examinerfour{Jurado externo o adicional}{Externo}{UNSA}
\dedicate{Escribe la dedicatoria}
```

Estos se llaman campos y sirven para indicar al documento la información particular de tu tesis. Entre cada pareja de símbolos `{ }` tienes que escribir el valor de ese campo. Por ejemplo en `\title{ }` va el título de tu tesis, en `\author{ }` tu nombre completo y así sucesivamente. Estos datos serán utilizados para construir la portada de tu tesis. Los campos `\examinerone{}` ... `\examinerfour{}` sirven para indicar los nombres de los miembros que integrarán al jurado en tu defensa de tesis. Generalmente son solo tres jurados, razón por la cual éste ultimo es opcional y su impresión esta en función de `\approved{}` como veremos en la siguiente sección

A.2. Generar de la tesis

En realidad lo anterior solo hemos cambiado el valor de las variables como ya lo hemos explicado, ahora recién comenzaremos a seleccionar lo que deseamos imprimir para nuestra tesis, más abajo podemos apreciar que con `\begin{document}` lo que hacemos es iniciar el documento y las dos respectivas carátulas o portadas. Estas no poseen paginación (obviamente).

```
\begin{document}
\makeFirstCover \makeSecondCover
```

A partir del `frontmatter` recién comenzamos a paginar con números romanos el contenido es muy intuitivo y fácil de darnos cuenta los campos que podemos varias, en el caso de `\approved{\tres}` nos representa que tenemos tres jurados y con `{\cuatro}`

obviamente si fuese el caso de un jurado adicional (aunque por ahora es poco usual puede darse el caso y seria más recomendable).

```
\begin{frontmatter}
\approved{\tres}%  {\tres} or {\cuatro}
\dedicatory
\begin{singlespace}
\tableofcontents \listoffigures \listoftables \pagebreak
\end{singlespace}
\myAcknowledgements{Agradecimientos}%
\myResumen{Resumen}%
\myAbstrac{Abstract}%
\end{frontmatter}
```

Los valores de `Agradecimientos` , `Resumen` , `Abstract` son nombres de archivos `.tex` externos que nos permiten tener ordenado el archivo principal.

A partir de aquí considero que ya tienes una idea muy clara sobre el manejo de la plantilla con el fin del ambiente `frontmatter` recien comienza la paginación normal, y el desarrollo de tu tesis, el índice general, de figuras y cuadros se genera automáticamente mediante los comandos indicados arriba (claramente en inglés *of course*).

Aquí puedes agregar tanto archivos externos como sea necesario mediante una simple linea, como por ejemplo: `\include{CapN}`, agrega el archivo `CapN.tex` al contenido total de la tesis, de manera análoga podes quitar un archivo si lo ves por conveniente, el formato bibliográfico utilizado en este caso es el de la ACM (*Association Computing Machinary*), existen muchos estilos disponibles en internet como de la IEEE, Harvard, etc. que puedes cambiar con solo modificar el valor del campo. en `\bibliographystyle{acm}` de `acm` por `ieee`, siempre y cuando dispongas de ese estilo en tu PC.

```
\pagestyle{fancyplain}
\include{Cap1}
\include{Cap2}
\include{Cap3}
\include{Cap4}
\include{Cap5}
\include{Cap6}
\myappendix{Apendice}
\begin{singlespace}
```

```
\bibliographystyle{acm}
\mybibliography{biblio}
\end{singlespace}
\end{document}
```

Toda la bibliografía que uses y referencies debe esta en `\mybibliography{biblio}` significa que el archivo destinado a esta labor es `biblio.bib`, con WinEdt existen macros que te permiten llenar de una forma muy sencilla los campos de una bibbliografia, por ejemplo:

```
@INPROCEEDINGS{Lerner00,
  AUTHOR =       {Barbara Staudt Lerner},
  TITLE =        {A Model for Compound Type Changes in Schemes},
  BOOKTITLE =    {ACM Transactions on Database Systems},
  YEAR =         {2000},
  volume =       {25},
  number =       {1},
  pages =        {83-127},
  month =        {March},
  organization = {ACM},}
```

Para referencias de autores es muy simple basta con colocar `\cite{Lerner00}` de esta forma el autor, previamente definido en el archivo biblio.bib quedara simplemente como un link: así [Lerner, 2000] y este es procesado automáticamente en la bibliografía(incluyendo su número u orden correlativo según le corresponda), más ejemplos son: [Abiteboul and Bonnerssss, 1991], [Batini et al., 1986], [Bertino, 1992], [Atkinson et al., 1989]

Apéndice B

Archivos Incluidos

Archivos del formato de tesis.

unsa.cls el formato general del documento de tesis.

Tesis.tex tendrá la estructura principal de tu tesis.

Agradecimientos.tex contenido de los agradecimientos

Resumen.tex contenido del resumen

Abstract.tex contenido del abstract.

Cap1.tex contenido del capítulo 1.

Cap2.tex contenido del capítulo 2.

Apendice.tex ejemplo de un apéndice.

biblio.bib ejemplos de referencias bibliográficas.

Logotipos de la UNSA.

escuela.eps logo EPIS en tonos de grises, formato eps.

logo.eps logo UNSA en tonos de grises, formato eps.

Bibliografía

[Abiteboul and Bonnerssss, 1991] Abiteboul, S. and Bonnerssss, A. (1991). Object and views. In *Proceedings of International Conference on Management of Data*, pages 238–247. ACM SIGMOD.

[Atkinson et al., 1989] Atkinson, M., Bancilhon, F., DeWitt, D., Dittich, K., Maier, D., and Zdonik, S. (1989). The object-oriented database system manifesto. In *Proceedings of International Conference on Deductive and Object-Oriented Databases*, pages 40–57. DOOD 89.

[Batini et al., 1986] Batini, C., Lenzerini, M., and Navathe, S. B. (1986). A comparative analysis of methodologies for database schema integration. In *ACM Computing Surveys*, volume 18, pages 321–364.

[Bertino, 1992] Bertino, E. (1992). A view mechanism for object-oriented databases. In *3rd International Conference on Extending Database Technology*, pages 136–151. EDBT 92.

[Deilis Carrazana Galán, 2015] Deilis Carrazana Galán, D. M. B. (2015). *Herramienta informática para la evaluación de la ambigüedad en textos legales*. PhD thesis, Universidad de las Ciencias Informáticas departamento Ciencias Informáticas, La Habana, Cuba.

[Lerner, 2000] Lerner, B. S. (2000). A model for compound type changes encountered in schema evolution. In *ACM Transactions on Database Systems*, volume 25, pages 83–127. ACM Inc.

[Lesk, 1986] Lesk, M. E. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *SIGDOC '86*.

[Torres-Ramos, 2012] Torres-Ramos, S. (2012). Estudio sobre métodos tipo lesk usados para la desambiguación de sentidos de palabras. *Research in Computing Science*, 47:139 – 158.