

**UNIVERSIDAD NACIONAL SAN AGUSTÍN DE AREQUIPA  
ESCUELA DE POSGRADO  
UNIDAD DE POSGRADO DE LA FACULTAD DE INGENIERÍA DE  
PRODUCCIÓN Y SERVICIOS**



**Método de Agrupamiento no Supervisado Para el  
Procesamiento del Lenguaje Natural Utilizando Medidas  
de Similitud Asimétricas y Propiedades Paradigmáticas**

Tesis presentada por:  
**MSc. Santisteban Pablo, Julio Omar**

Para optar el Grado de:  
**Doctor en Ciencias de la Computación**

Asesor:  
**Dr. Javier Leandro Tejada Cárcamo**

Arequipa - Perú  
2016



No existe inspiración tan hermosa que el del ser amado, quien con palabras de amor y ternura me llevo de la mano en esta investigación y en cada paso de mi vida, Fabiola eres mi musa y mi oxígeno de vida, agradezco a Dios por bendecir nuestro Amor, estas letras son más tuyas que mías, por siempre te amare.

*Antes, ahora y siempre te Amare*

**Fabiola Muñoz Velando.**

La motivación más grande es de mis dos hermosas hijas  
**Jennifer y Alison Santisteban Muñoz.**

Sin duda alguna el apoyo moral de mi familia y mi familia política (Jaime, Nilda, Soledad, Verónica, Janet, Carlos, Angelita, Víctor, Abel, Alejandra, Patricio, Mauricio, Ricardo, Eiki) me ha permitido forjar cada línea.

Cada algoritmo no se podría haber forjado sin el apoyo científico y técnico de mis colegas de la Universidad Católica San Pablo y de Qantas AirWay.



## **Declaración**

Por la presente declaro que, salvo cuando se hace referencia específica a la labor de otros, el contenido de esta tesis doctoral es originales y no han sido presentado en su totalidad o en parte, para su consideración por cualquier otro título o calificación en esto, o cualquier otra universidad. Esta tesis es mi propio trabajar y no contiene nada que es el resultado del trabajo realizado en colaboración con los demás, excepto como se especifica en el texto y Reconocimientos.

MSc. Santisteban Pablo, Julio Omar  
Abril 2021



## **Reconocimiento**

Este trabajo de investigación ha sido financiado por el Fondo para la Innovación, Ciencia y Tecnología - Peru (FINCyT).

Agradezco la guía y apoyo incondicional que mi asesor Dr. Javier Leandro Tejada Cárcamo ha tendido connmigo a lo largo de esta travesía científica.

Agradezco el profundo soporte moral y científico que he recibido de la Universidad Católica San Pablo, en particular del Dr. Gonzalo Fernández del Carpio y del Dr. Ernesto Cuadros Vargas, agradezco profundamente a mis colegas que en cada conversación me han inspirado.



## Resumen

Una de las tareas más comunes para el ser humano, pero de con una alta complejidad es la agrupación y clasificación. Por otro lado, la debilidad del ser humano es la capacidad de procesar altas cantidades de datos y de forma rápida, característica propia de los computadores. Hoy en día se generan grandes cantidades de datos en el Internet, datos de distintos tipos y con diferentes objetivos. Para esto se necesitan de algoritmos de agrupación que nos permitan identificar los distintos grupos y características de estos grupos, de forma automática sin conocimiento previo. Por otro lado, es importante definir con claridad qué medida de similitud se utilizará en el proceso de agrupación, la gran mayoría de las medidas de agrupación se enfocan en un aspecto simétrico.

En la presente tesis se propone una novedosa medida de similitud asimétrica, Coeficiente d Similitud Unilateral Jaccard (uJaccard), similitud no es igual entre dos objetos  $uJaccard(a,b) \neq uJaccard(b,a)$ . Así también se presenta una similitud asimétrica con pesos Coeficiente Ponderado de Similitud Unilateral Jaccard, la cual mide el nivel de incertidumbre entre dos objetos. Así también en esta tesis se propone una nueva propiedad de grafos, la propiedad paradigmática la cual considera la equivalencia regular como característica fundamental y por último se propone un algoritmo de agrupación PaC, por sus siglas en inglés Paradigmatic Clustering, el cual incorpora la uJaccard y la propiedad paradigmática. Se ha realizado evaluaciones extensivas con datos pequeños, reales, sintéticos y se ha procesado 3 grandes corpus.

Se ha demostrado que PaC es un algoritmo que supera los resultados de algoritmos de agrupación del estado del arte. Más aun PaC es un algoritmo capaz de ser ejecutado de forma paralela, distribuida, incremental y en flujo, características que se necesitan para el procedimiento de grandes cantidades de datos y de constante generación de datos.



## **Lista de publicaciones originales**

- Santisteban, J., Tejada Cárcamo, J. (2015). "Unilateral Jaccard Similarity Coefficient". SIGIR (Special Interest Group in Information Retrieval) Workshop on Graph Search and Beyond.
- Santisteban, J., Tejada Cárcamo, J. (October-2015). "Unilateral Weighted Jaccard Coefficient for NLP". IEEE Mexican International Conference in Artificial Conference.
- Santisteban, J., Tejada Cárcamo, J. (November-2015). "Paradigmatic Clustering for NLP". IEEE International Conference on Data Mining Workshops, ICDMW.
- Santisteban, J., Tejada Cárcamo, J. (2016). "Clustering algorithm based on asymmetric similarity and paradigmatic features". IJICA International Journal of Innovative Computing & Applications. Vol 7 Num 4.
- Santisteban, J. (Año 2012). "Método De Clusterización De Textos Basado En La Semántica Ontológica De La Lengua Inglesa". Universidad Católica San Pablo, Revista De Investigación Nº 3, Volúmen 3. ISSN Versión Impresa 2309-6683.



## **Contribución del Autor al Estado del Arte**

El presente trabajo de investigación doctoral tiene 4 aportes al estado del arte:

1. Nuevo Coeficiente de Similitud Unilateral Jaccard, el cual es una extensión del coeficiente de Jaccard e incorpora el concepto de similitud asimétrica entre dos objetos.
2. Nueva propiedad Paradigmática para grafos, el cual es definido como el conjunto de vértices que son similares si ellos tienen una equivalencia regular y son intercambiables por sustitución o transposición.
3. Nuevo algoritmo de agrupación PaC (Paradigmatic Clustering) para grafos.
4. Nuevo Coeficiente Ponderado de Similitud Unilateral Jaccard, el cual es un coeficiente de incertidumbre e incorpora el concepto de incertidumbre asimétrica entre dos objetos.



# Índice General

<b>Lista de Figuras</b>	<b>7</b>
<b>Lista de Tablas</b>	<b>9</b>
<b>1 Introducción</b>	<b>1</b>
1.1 Motivación . . . . .	1
1.2 Planteamiento del Problema . . . . .	1
1.3 Objetivo . . . . .	3
1.3.1 Objetivo General . . . . .	3
1.3.2 Objetivo Específicos . . . . .	3
1.4 Estructura de la tesis . . . . .	3
<b>2 Similitud y Distancia</b>	<b>5</b>
2.1 Introducción . . . . .	5
2.2 Definición . . . . .	6
2.3 Similitud y distancia para variables binarias . . . . .	6
2.3.1 Distancia Hamming . . . . .	7
2.3.2 Distancia simple de igualdad . . . . .	7
2.3.3 Distancia de Jaccard . . . . .	7
2.4 Distancia para variables Nominales . . . . .	8
2.4.1 Representando cada categoría como una variable binaria . . . . .	9
2.4.2 Representando cada categoría por una serie de variables binarias . . . . .	9
2.5 Distancia para variables Ordinales . . . . .	10
2.5.1 La Transformación Normalizada de rango . . . . .	10
2.5.2 Distancia de Spearman . . . . .	11
2.5.3 Distancia de Footrule . . . . .	11
2.6 Distancia para variables cuantitativas . . . . .	12
2.6.1 Distancia Euclíadiana . . . . .	12

2.6.2	Distancia de Minkowski de orden $\lambda$ . . . . .	12
2.6.3	Distancia de ciudad (City block distance) . . . . .	13
2.6.4	Distancia de Chebyshev . . . . .	13
2.6.5	Distancia de Canberra . . . . .	13
2.6.6	Distancia de Bray-Curtis . . . . .	13
2.6.7	Separación Angular . . . . .	14
2.6.8	Coeficiente de Correlación . . . . .	14
2.6.9	Distancia de Mahalanobis . . . . .	14
2.7	Medidas de Similitud Semánticas . . . . .	15
2.8	Medidas basadas en el largo del camino . . . . .	15
2.8.1	Camino más corto . . . . .	15
2.8.2	Medida de Wu & Palmer . . . . .	16
2.8.3	Medida de Leacock & Chodorow . . . . .	16
2.8.4	Medida de Li . . . . .	17
2.9	Medidas basadas en el contenido de la información . . . . .	17
2.9.1	Medida de Resnik . . . . .	17
2.9.2	Medida de Lin . . . . .	18
2.9.3	Medidad de Jiang . . . . .	18
2.10	Medidas basadas en características . . . . .	18
2.10.1	Similitud Asimétrica de Tversky . . . . .	19
<b>3</b>	<b>Métodos de Agrupamiento No Supervisado</b>	<b>21</b>
3.1	Introducción . . . . .	21
3.2	Definición de Agrupamiento no Supervisado . . . . .	22
3.2.1	Técnicas de Agrupamiento no Supervisado . . . . .	22
3.3	Técnicas de particionamiento . . . . .	23
3.3.1	K-Means Clustering . . . . .	23
3.3.2	K-Medoid Clustering . . . . .	24
3.3.3	Fuzzy C-Means Clustering . . . . .	25
3.4	Agrupación basado en la distribución . . . . .	26
3.5	Las técnicas de agrupamiento jerárquico . . . . .	27
3.6	Agrupamiento basado en densidad . . . . .	28
3.7	Agrupamiento basado en mallas . . . . .	29
3.8	Mecanismo de división y combinación . . . . .	30
3.9	Medidas de validación de agrupación . . . . .	31
3.10	Agrupación no Supervisada Basado en Grafos . . . . .	33
3.11	Medidas de Calidad y Evaluación para Grafos . . . . .	35

<b>4 Agrupación Paradigmática</b>	<b>39</b>
4.1 Introducción . . . . .	39
4.1.1 Agrupamiento . . . . .	41
4.2 Definición de Similitud Paradigmática . . . . .	42
4.2.1 Conceptos Básicos de las Estructuras Paradigmáticas . . . . .	42
4.2.2 Extended Similitud Paradigmática . . . . .	42
4.3 Evaluacion Experimental de uJaccard . . . . .	44
4.3.1 Pruebas Pequeñas . . . . .	44
4.3.2 Cortando Grafos . . . . .	45
4.3.3 Redes Sociales . . . . .	46
4.4 Agrupación de grafos usando la estructura paradigmática . . . . .	49
4.4.1 Estructura Paradigmatica . . . . .	51
4.4.2 Similitud Paradigmatic . . . . .	53
4.4.3 Algoritmo . . . . .	53
4.4.4 PaC algoritmo . . . . .	53
4.5 PaC Evaluación Experimental . . . . .	55
4.5.1 Evaluación con grafos pequeños . . . . .	55
4.5.2 Grafos reales . . . . .	56
4.5.3 Datos Sintéticos . . . . .	57
4.5.4 Evaluación NLP . . . . .	58
4.6 Conclusion . . . . .	65
<b>5 Coeficiente Ponderado de Similitud Unilateral Jaccard</b>	<b>67</b>
5.1 Introducción . . . . .	67
5.2 Coeficiente Ponderado de Similitud Unilateral Jaccard . . . . .	67
5.2.1 Construcción del Grafo de Palabras . . . . .	73
5.2.2 Incertidumbre de “man bites dog” y “dog bites man” . . . . .	73
5.3 Conclusion . . . . .	73
<b>6 Conclusiones</b>	<b>77</b>
6.1 Introducción . . . . .	77
6.2 Conclusiones . . . . .	77
6.3 Trabajos Futuros . . . . .	78
<b>Bibliografía</b>	<b>79</b>
<b>Apendice A Artículo I</b>	<b>87</b>

---

**Índice General** **6**

<b>Apendice B Artítulo II</b>	<b>93</b>
<b>Apendice C Artículo III</b>	<b>101</b>
<b>Apendice D Artículo IV</b>	<b>109</b>
<b>Apendice E Artículo V</b>	<b>125</b>

# **Lista de Figuras**

4.1	Equivalencia Estructural . . . . .	42
4.2	Simple graph . . . . .	44
4.3	Toy graph. . . . .	44
4.4	Grafo Pequeño. . . . .	47
4.5	Grafo Pequeño. . . . .	48
4.6	Coautoría rede de científicos. . . . .	49
4.7	Comparación de PaC en datos sintéticos, N=5000 . . . . .	58
4.8	Comparación de PaC en datos sintéticos, N=1000 . . . . .	59
4.9	Comparación de PaC en datos sintéticos, N=1000, comunidades grandes . .	59
4.10	Plabra "chair" de en.wikipedia . . . . .	62
4.11	Palabra "chair" de BNC. . . . .	63
4.12	Palabra "sail" de BNC . . . . .	63
4.13	Palabra "god" de BNC . . . . .	64
5.1	Aristas entre palabras $W_1$ y $W_2$ con pesos. . . . .	69
5.2	Aristas entre palabras $W_1$ y $W_2$ con múltiples caminos . . . . .	71



# Listas de Tablas

2.1	Matriz de confusión . . . . .	7
2.2	Caso de desempeño . . . . .	11
3.1	Algoritmos de agrupamiento en grafos. . . . .	36
3.2	Medida de Evaluación Zheng Chen et al [CJ10] . . . . .	37
4.1	Calculos de uJaccard de la figura 4.3 . . . . .	45
4.2	pruebas de uJaccard en pequeños grafos . . . . .	46
4.3	Cortando el grafo (figura 4.4) usando uJaccard . . . . .	47
4.4	Cortando el grafo (figura 4.5) usando uJaccard . . . . .	48
4.5	Calculando uJaccard de la figure 4.5, búsqueda paradigmática. . . . .	50
4.6	Similitud de uJaccard entre mejores 100 y 250 actores y actrices. . . . .	51
4.7	Similitud de uJaccard de los mejores 250 actores y actrices en niveles. . . . .	52
4.8	Agrupacion de pequeños grafos utilizando PaC . . . . .	56
4.9	Agrupación de redes reales utilizando PaC . . . . .	57
4.10	Tamaños de los corpus . . . . .	60
4.11	Grupos o sentidos de la palab "apple" . . . . .	64
4.12	Grupos o sentidos de la palabra "god" . . . . .	65
5.1	Vector de características Normalizadas de una palabra $W_i$ . . . . .	68
5.2	uwJaccard ( $W_1, W_2$ ) . . . . .	70
5.3	uwJaccard ( $W_1, W_2$ ) con diferentes caminos. . . . .	72
5.4	Incertibumbre entre <i>man</i> y <i>dog</i> . . . . .	74
5.5	" <i>man bites dog</i> " o " <i>dog bites man</i> " . . . . .	74



# **Capítulo 1**

## **Introducción**

---

### **1.1 Motivación**

Una de las tareas más comunes para el ser humano, pero con una alta complejidad es la agrupación y clasificación. Por otro lado, la debilidad del ser humano es la capacidad de procesar altas cantidades de datos y de forma rápida, característica propia de los computadores. Hoy en día se generan grandes cantidades de datos en el Internet, datos de distintos tipos y con diferentes objetivos. Así también existen grandes conjuntos de datos como el Genoma Humano con un tamaño de 3,200 millones de pares.

La agrupación es una herramienta importante en la minería de datos y aprendizaje automático. Su objetivo es la partición de los objetos de un conjunto de datos para que los objetos similares se pongan en un mismo grupo y diferentes objetos en diferentes grupos.

### **1.2 Planteamiento del Problema**

La clasificación de un conjunto de objetos en formar grupos, es un problema de análisis de datos antiguo. Cuyo escenario más común es en el que los datos tienen propiedades con distintas medidas. Esta naturaleza de alta dimensión de clasificación proporciona una oportunidad, pero también presenta algunas dificultades para el desarrollo de algoritmos que ataquen este problema.

Se pueden distinguir dos grandes categorías de problemas de clasificación. En el primero, a priori se tiene las definiciones de las clases a las que pertenece los datos, esto quiere decir que se conocen los grupos o clases y las características de cada grupo o clase y la tarea a

desarrollar es la asignación de los datos a los grupos o clases, esto se puede determinar a través del análisis de los datos. En la literatura este tipo de tarea se denomina agrupación de aprendizaje no supervisado o simplemente clasificación.

En algunas tareas no se conoce a priori los grupos o clases, tampoco las características de ellas o si el número de grupo existen, la asignación se da a través del análisis de los datos, identificando las características de los datos y agrupándolos según sus características, formando grupos según las características en los datos. En la literatura este tipo de tarea se denomina agrupación de aprendizaje no supervisado o análisis de conglomerados o simplemente agrupación o en inglés *clustering*.

La agrupación (o análisis de conglomerados) es uno de los problemas más comunes en el reconocimiento de patrones y aprendizaje automático. El objetivo, sin ningún conocimiento a priori, es dividir un conjunto de datos en diferentes grupos de modo que los puntos de datos dentro de un grupo son similares y disimilares a los otros grupos. La agrupación es un problema NP-hard [cluster NP 1]. Esto significa que una solución de agrupación óptima no se puede lograr a excepción de muy pequeños conjuntos de datos.

La agrupación no supervisada es una técnica común para el análisis de datos. Se busca en los datos patrones suficientemente comunes, similares y a ser identificados de manera intuitiva como parte de un grupo común, este análisis debe observar diferentes patrones y realizar distintas observaciones con el fin de identificar las características propias de los grupos a formar, de tal manera que:

1. Cada grupo es homogéneo o compacto con respecto a ciertas características; es decir, las observaciones o patrones en cada grupo son similares entre sí.
2. Cada grupo debe ser diferente de los otros grupos con respecto a las mismas características; es decir, las observaciones de un grupo deben ser diferentes de las observaciones o patrones de otros grupos.

El primer paso en el proceso de agrupación es seleccionar una medida de la similitud (o distancia), pero la definición de similitud u homogeneidad varía de análisis a análisis y depende de los objetivos del estudio. La gran mayoría de los algoritmos de agrupación se basan en medidas simétricas.

Una vez que se selecciona una medida de distancia, los elementos en el conjunto de datos se pueden combinar y luego se toma una decisión sobre el tipo de técnica de agrupación para ser utilizado y en el método de agrupamiento particular. En general, los algoritmos de agrupamiento de datos pueden ser jerárquica o no jerárquica. Cada algoritmo se enfoca no el tipo de datos y el área de investigación como biología, imágenes, procesamiento del lenguaje natural entre algunas.

Por tanto, uno de los problemas que observamos es el escaso uso de medidas de similitud asimétricas, así como utilizar propiedades de agrupación particulares para el área de procesamiento del lenguaje natural.

La gran cantidad de datos cambia el paradigma de definición de algoritmos que puedan ser usados con miles de millones de datos, más aún la generación de datos es tan grande diariamente que se requiere de algoritmos de agrupación que puedan asignar los datos de forma fluida, así también la capacidad computacional y las redes nos permite interconectar servidores creando un algoritmo que puedan trabajar en un modelo distribuido.

## 1.3 Objetivo

### 1.3.1 Objetivo General

Desarrollar un método de agrupamiento no supervisado para el procesamiento del lenguaje natural utilizando medidas de similitud asimétricas y propiedades paradigmáticas.

### 1.3.2 Objetivo Específicos

1. Coeficiente de similitud unilateral, el cual incorpore el concepto de similitud asimétrica entre dos objetos.
2. Propiedad paradigmática para grafos, el cual se base en la equivalencia regular y los objetos sean intercambiables por sustitución o transposición.
3. Algoritmo de agrupación para grafos basado en una similitud unilateral asimétrica y la propiedad paradigmática.
4. Coeficiente ponderado de similitud unilateral asimétrico, basado en incertidumbre y soportado por la entropía de la información.

## 1.4 Estructura de la tesis

El resto de la tesis se organiza de la siguiente manera: En el Capítulo 2 se representa el estado del arte y marco teórico de las medidas de similitud y distancia, en el capítulo 3 se presenta el estado del arte y marco teórico de los métodos de agrupación, en el capítulo 4 se discuten y se introduce una nueva algoritmo de agrupación, en el capítulo 5 se discuten y se introduce un nuevo coeficiente de similitud asimétrico, en el capítulo 5 se presentan las conclusiones y en los apéndices se presentan los artículos publicados.



# **Capítulo 2**

## **Similitud y Distancia**

---

### **2.1 Introducción**

Los conceptos de similitud y distancia son cruciales en muchas áreas de las ciencias, medidas de similitud y distancia son usadas para calcular la similitud y/o distancia entre diferentes objetos, siendo este un requerimiento esencial en la mayoría de los algoritmos de reconocimiento de patrones, así como en los de agrupamiento, clasificación, selección de características, detección de perfiles, búsquedas. Existe una gran cantidad de medidas de similitud en la literatura científica, la selección de una medida de similitud apropiada para una tarea es crucial para la obtención de resultados malos, buenos o excelentes. La selección de una medida de similitud depende del tipo de objetos a usar y su comportamiento en el contexto. Este capítulo presenta las medidas de similitud más usadas en la literatura científica.

Similitud y la disimilitud entre objetos puede ser medido basado en varias características. En consecuencia la selección de características es requerida antes que se calcule la similitud y/o distancia entre objetos. Es así que la similitud y/o distancia entre objetos varía dependiendo del valor de las características seleccionadas. Un enfoque es la agregación de la similitud y/o distancia de cada característica y realizar una ponderación [BC57] [Eve78] [Mar96] [Tek06] [Web03].

Las medidas de similitud y/o distancia son medidas simétricas. Las medidas de distancia se consideran como el tamaño del espacio/brecha entre dos objetos en el espacio. La medida de similitud es el grado de semejanza de sus características. Por otro lado, se tiene que considerar la similitud, pero desde un punto de vista asimétrico, donde la similitud se da unilateral. Este punto lo veremos en el capítulo 4.

## 2.2 Definición

Para poder definir la similitud o disimilitud primero se tiene que definir el tipo de característica a usar. Es así que la relación que existe entre diferentes características de distintos objetos es usada para calcular la similitud entre los objetos. La similitud principalmente cuantifica la relación entre diferentes características.

Dados dos objetos  $i$  y  $j$ , la similitud entre estos dos objetos  $i$  y  $j$  se denota por  $S_{ij}$ . El valor de  $S_{ij}$  depende en gran medida de la escala de las mediciones y también de los tipos de datos de las características seleccionadas. Por otro lado, la distancia o disimilitud mide la diferencia entre dos puntos, basado en los valores de sus atributos. La distancia entre dos objetos  $i$  y  $j$  se denota por  $d_{ij}$ .

La distancia es un valor cuantitativo el cual debe de satisfacer la siguiente condición  $d_{ij} \geq 0; d_{ii} = 0; d_{ij} = d_{ji}$  (*simétrica*);  $d_{ij} \leq d_{ik} + d_{jk}$  (debe de satisfacer la desigualdad triangular).

La medida de similitud entre dos objetos es importante para distinguir las diferencias entre dos objetos. Así también los objetos deben de distinguirse del uno al otro. Cualquier algoritmo podrá agrupar los objetos según sus medidas de similitud. Las medidas de similitud podrán proveer información de las características comunes de los grupos formados. Los algoritmos podrán agrupar los objetos en distintas formas según las medidas de similitud de sus distintas características [Tek06].

Las medidas de similitud son esenciales para los algoritmos no supervisados, así como para los algoritmos supervisados, clasificadores. Las medidas de similitud pueden ser usadas para realizar previsiones en el comportamiento de los objetos. Los grupos pueden ser explorados basados en las medidas de similitud, identificando objetos más o menos similares. La minería de datos puede utilizar medidas de similitud de distintas selecciones de características. La toma de decisión y planteamiento llega a ser mucho más clara cuando se conoce la estructura de datos y su comportamiento el cual es a través de las medidas de similitud.

## 2.3 Similitud y distancia para variables binarias

Variables binarias solo pueden optar los valores 0 o 1, verdadero o falso, positivo o negativo. Para medir la similitud o distancia entre dos objetos, estos objetos deben ser representados por un vector de valores binarios [Tek06].

Dados dos objetos  $i$  y  $j$ , con 3 características las cuales están parametrizadas como sigue, característica  $A$  es 1 si el valor es  $>$  a 0.5, característica  $B$  es 1 si el valor es  $>$

Tabla 2.1 Matriz de confusión

Objeto i \ Objeto j	0	1
0	P00 = 1	P01 = 2
1	P10 = 0	P11 = 1

a 13, característica  $C$  es 1 si el valor es verdadero; caso contrario el valor es 0. Siendo los objetos  $i, j$  tridimensionales representado por tres características con valores como sigue:  $i = (0.53, 4, \text{verdadero})$  y  $j = (0.25, 12.5, \text{falso})$ , aplicando los parámetros obtenemos  $i = (1, 0, 1)$  y  $j = (0, 0, 0)$ . Aplicando la matriz de confusión podemos decir que la medida de similitud o distancia es la suma de las instancias de sus combinatorias  $d_{ij}os_{ij} = 1 + 0 + 2 + 0$  [BS12].

### 2.3.1 Distancia Hamming

La distancia Hamming entre dos objetos representados por sus vectores binarios, es igual al número de posiciones donde estos tienen distintos dígitos y se define como: [Ham50][Tan05][Tek06]  $d_{ij} = P01 + P10$ . Siguiendo el ejemplo anterior la distancia Hamming es 2.

### 2.3.2 Distancia simple de igualdad

La función de distancia es usada para calcular la disimilitud entre dos objetos binarios cuando la frecuencia de ocurrencia de 0 y 1 son iguales en los dos objetos, el cual es definido como:  $d_{ij} = (P01 + P10)/F$ . Donde  $F$  es el número de características. Siguiendo el ejemplo anterior  $d_{ij} = (0 + 2)/3 = 2/3 = 0.67$ .

### 2.3.3 Distancia de Jaccard

El coeficiente de Jaccard es una medida de similitud, mientras que la distancia de Jaccard es una distancia de medida. Estas medidas se utilizan particularmente cuando los valores binarios 0 y 1 no tienen la misma frecuencia de ocurrencia. Por ejemplo, en un supermercado existen muchos productos y los clientes compran distintos productos. Si se desea averiguar el número de productos comunes que los clientes compran, podríamos adoptar dos enfoques. El primero identificando los productos no comunes, lo cual se puede lograr con la “Distancia simple de igualdad”; el segundo enfoque, es más simple, es calcular el coeficiente de Jaccard el cual se define como:

$$S_{ij} = P_{11}/(P_{11} + P_{01} + P_{10}). \quad (2.1)$$

La distancia de Jaccard es definida como:

$$D_{ij} = 1 - S_{ij} = (P_{01} + P_{10}) / (P_{01} + P_{10} + P_{00}). \quad (2.2)$$

Para el ejemplo anterior el coeficiente de Jaccard es de 1/3 y la distancia de Jaccard es de 2/3. Es así que el coeficiente de Jaccard se puede generalizar para ser aplicado a objetos con características no binarias, esto se logra utilizando teoría de conjuntos. Si Tenemos dos conjuntos A y B en coeficiente de Jaccard se define como:

$$S_{ij} = |AnB| / |AuB|. \quad (2.3)$$

Si tenemos los siguientes conjuntos  $A = \{6, 7, 8, 9\}$ ,  $B = \{7, 8, 9, 10\}$ , entonces  $AuB = \{6, 7, 8, 9, 10\}$  y  $AnB = \{7, 8, 9\}$ , entonces  $SAB = 3/5 = 0.6$ .

## 2.4 Distancia para variables Nominales

Las variables categóricas nominales tienen sus valores representados por categorías que no obedecen a una clasificación intrínseca y no pueden ser medidos cuantitativamente. Los valores numéricos son usados para representar las distintas categorías. Por ejemplo, 55 = animal, 42 = vegetal; código postal: 2000 = Sydney, 2150 = Parramatta, 2170 = Liverpool, 2076 = Wahroonga ... La etiqueta numérica no representa en absoluto un orden, más aún los valores de las etiquetas pueden cambiar sin afectar la representación lógica y cálculos de ninguna forma.

Para calcular la distancia entre dos objetos que tengan características nominales, primero se debe de identificar el número de categorías, en el caso de existir una sola categoría la distancia es 0 o no existe, en el caso de existir dos categorías se puede usar las medidas similitud y distancia para variables binarias como Jaccard, Hamming entre otros. Si existen más de dos categorías se debe de transformar las  $n$  categorías a un modelo binario, representando cada categoría como una variable binaria o representando cada categoría por una serie de variables binarias.

La distancia entre dos objetos con características nominales es calculada como la taza de desigualdades sobre el total número de variables binarias. El cual se define como:

$$D_{ij} = (q + p) / t. \quad (2.4)$$

Donde  $q$  es el número de variables  $P_{01}$ , y  $p$  es el número de variables  $P_{10}$ , y  $t$  es el número total de variables binarias utilizadas para la representación de la variable nominal.

### 2.4.1 Representando cada categoría como una variable binaria

La distancia entre dos objetos nominales es la taza de disimilitud sobre el número total de variables binarias, para este caso se usa la distancia Hamming.

Por ejemplo, tres objetos tienen dos variables nominales género el cual tiene dos valores 0 = hombre y 1 = mujer y la segunda variable nominal es profesión el cual tiene cuatro opciones abogado, ingeniero, médico, profesor. Los tres objetos con sus respectivos valores Objeto1 = (0,(1,0,0,0)), Objeto2 = (1,(0,1,0,0)) y Objeto3 = (1,(0,0,0,1)). Para calcular las distancias entre los objetos se utilizará la distancia Hamming como sigue:

Distancia (Obj1 , Obj2) es de (1,2) y su distancia en conjunto es 3.

Distancia (Obj1 , Obj3) es de (1,2) y su distancia en conjunto es 3.

Distancia (Obj2 , Obj3) es de (0,2) y su distancia en conjunto es 2.

La distancia entre dos objetos nominales se calcula a través de la taza de disimilitud sobre el número total de variables binarias.

Distancia (Obj1 , Obj2) es de (1/1,2/4) y la distancias media es  $(1+2/4)/2 = 0.75$ .

Distancia (Obj1 , Obj3) es de (1/1,2/4) y la distancias media es  $(1+2/4)/2 = 0.75$ .

Distancia (Obj2 , Obj3) es de (0/1,2/4) y la distancias media es  $(0+2/4)/2 = 0.25$ .

### 2.4.2 Representando cada categoría por una serie de variables binarias

Para poder representar n variables nominales en un modelo binario se utiliza la siguiente formula  $c < 2^{du}$ ;  $du = |||logc / log2|||$ . Donde c es el número de clases y du es el número de variables binarias que representaran las variables nominales.

Por ejemplo para representar las profesiones en el ejemplo anterior  $du = |||log4 / log2||| = |||2.0||| = 2$ . Los cuales se representan de la siguiente forma 00 como abogado, 01 como ingeniero, 10 como médico, 11 como profesor. Siguiendo el ejemplo anterior el Objeto1 (0,(0,0)), Objeto2 (1,(0,1)), Objeto3 (1,(1,1)). Para calcular la distancia entre variables nominales primero se tiene que calcular la distancia Hamming, luego se calcula la taza de disimilitud sobre el número total de variables binarias.

Distancia (Obj1 , Obj2) es de (1,1) y su distancia en conjunto es 2.

Distancia (Obj1 , Obj3) es de (1,2) y su distancia en conjunto es 3.

Distancia (Obj2 , Obj3) es de (0,1) y su distancia en conjunto es 1.

La distancia entre dos objetos nominales se calcula a través de la taza de disimilitud sobre el número total de variables binarias.

Distancia (Obj1 , Obj2) es de  $(1/1,1/4)$  y la distancias media es  $(1+1/4)/2 = 0.625$ .

Distancia (Obj1 , Obj3) es de  $(1/1,2/4)$  y la distancias media es  $(1+2/4)/2 = 0.375$ .

Distancia (Obj2 , Obj3) es de  $(0/1,1/4)$  y la distancias media es  $(0+1/4)/2 = 0.125$ .

## 2.5 Distancia para variables Ordinales

Las variables ordinales presentan un orden que es dado por el grado de representación de cada variable, las variables ordinales son utilizadas para indicar un relativo orden o ranking de algunos objetos [HD79] [Tek06] [BS12]. Es así que el valor cuantitativo entre dos variables ordinales no es de importancia, pero el orden en el que están ubicados estas variables son de importancia. Por ejemplo, consideremos el orden de los grados académicos, primaria, secundaria, técnico, bachiller, Magister, Doctor o el grado sobresaliente de un estudiante en un curso, desaprobado, aprobado, notable, sobresaliente, excelencia. Las variables pueden ser expresadas como letras AA, AB, AC o como números ascendentes o descendientes, según sea el caso más apropiado.

Para poder calcular la distancia o disimilitud entre dos variables ordinales se pueden usar una serie de métodos entre ellos La Transformación Normalizada de rango, distancia de Spearman, distancia de footrule, distancia de Kendall, distancia de Cayley, distancia de Hamming, distancia de Chebyshev/Maximum,distancia de Minkowski, entre otros.

### 2.5.1 La Transformación Normalizada de rango

Para calcular la distancia, se cuantifica las variables ordinales y se normalizan, una vez cuantificada se pueden usar medidas cuantitativas de distancia. Las variables ordinales se tienen que evaluar si se pueden cuantificar, por ejemplo, en el caso de grados académicos no podríamos cuantificar el grado de doctor en comparación a un bachiller. Por otro lado, variables ordinales como grados sobresalientes en un curso, sí se pueden cuantificar. Para aquellas variables que no se puedan cuantificar se pueden utilizar otras medidas como distancia de Spearman, distancia de Kendall, distancia de Cayley, distancia de Hamming , distancia de Ulam o distancia de Chebyshev/Maximum entre otras [HD79] [RCP00] [Tek06] [BS12].

Para hallar la distancia entre dos variables ordinales se deben de convertir a un rango de  $r = 1$ , (menor rango) hasta  $R$ , (mayor rango). Luego el rango es normalizado entre una

Tabla 2.2 Caso de desempeño

( objeto )	curso 1	curso 2	curso 3	curso 4	curso 5	curso 6
<b>Alumno 1</b>	aprobado	notable	sobresaliente	aprobado	notable	excelencia
<b>Alumno 2</b>	sobresaliente	excelencia	aprobado	notable	aprobado	sobresaliente

escala [0,1] por medio de la siguiente formula  $x = r - 1/R - 1$ . Y posteriormente se puede aplicar cualquier medida cuantitativa.

Por ejemplo, si se quiere comparar dos alumnos según su grado de sobresaliente en 6 cursos llevados en un semestre.

La conversión de los grados de sobresaliente a rango es de la siguiente forma 1 = desaprobado, 2 = aprobado, 3 = notable, 4 = sobresaliente, 5 = excelencia. Luego el rango es normalizado a una escala [0,1] de la siguiente forma: *desaprobado* =  $1 - 1/5 - 1 = 0/4 = 0$ , *aprobado* =  $2 - 1/5 - 1 = 1/4$ , *notable* =  $3 - 1/5 - 1 = 2/4$ , *sobresaliente* =  $4 - 1/5 - 1 = 3/4$ , *excelencia* =  $5 - 1/5 - 1 = 4/4 = 1$ . Para hallar la distancia se puede usar cualquier medida cuantitativa de distancia, por ejemplo, distancia de *Euclides* = 1.391941, distancia de *Coseno* = 0.435218.

### 2.5.2 Distancia de Spearman

La distancia de Sperman se calcula como el cuadrado de la distancia de Euclides de dos vectores ordinarios de la siguiente forma [Tek06] [RCP00] [HD79] [BS12] :

$$d_{ij} = \sum_{p=1}^n (x_{ip} - x_{jp})^2. \quad (2.5)$$

Donde n es el número de características ordinales. Siguiendo el ejemplo anterior, la distancia de Sperman entre alumno1 y el alumno2 = 1.937

### 2.5.3 Distancia de Footrule

La distancia de Footrule es la suma absoluta de las diferencias entre los valores de las características de dos vectores ordinales. La ecuación de esta es muy similar al de la distancia de City Block o distancia de Manhattan, otro nombre que se le da es de distancia Sperman Footrule. La distancia Footrule es definida como [Tek06] [RCP00] [HD79] [BS12]:

$$d_{ij} = \sum_{k=1}^n |x_{ik} - x_{jk}|. \quad (2.6)$$

Siguiendo el ejemplo anterior, la distancia de footrule entre alumno1 y alumno2 = 3.25

## 2.6 Distancia para variables cuantitativas

Las variables cuantitativas representan valores escalares que representan una magnitud en particular, se pueden medir en una escala numérica. Por ejemplo, peso, altura, edad, salario, temperatura, área, etc. Un objeto puede ser representado por una serie de características cuantitativas, cada variable cuantitativa constituye una dimensión, por ejemplo, un objeto automóvil puede ser representado por velocidad máxima, peso neto, numero de válvulas, cilindraje; cada una de las características representa un punto en un modelo de 4 dimensiones. [Tek06] [BS12]

### 2.6.1 Distancia Euclidiana

La distancia más comúnmente usada para medir variables cuantitativas [PLS04] [Tek06]. En términos generales cuando se habla de distancia, se está hablando de la distancia Euclidiana. Esta mide la raíz cuadrada de la diferencia entre un par de coordenadas de un objeto en un plano. La distancia Euclidiana es un caso especial de la distancia de Minkowski con un  $\lambda = 2$ , la fórmula es dada como:

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}. \quad (2.7)$$

### 2.6.2 Distancia de Minkowski de orden $\lambda$

Es una medida de distancia general, esta puede ser usada para variables ordinales y quantitativas, la cual depende del valor de  $\lambda$ , esta distancia es igual a otras distancias, esto es según el uso que se le dé al valor  $\lambda$ , cuando  $\lambda = 1$  entonces la distancia Minkowski es la distancia de Geometría del taxista (City block distance or taxicab geometry), cuando  $\lambda = 2$  entonces la distancia Minkowski es la distancia Euclidiana, cuando  $\lambda = \infty$  entonces la distancia Minkowski es la distancia de Chebyshev [PLS04] [Tek06], la fórmula es dada como:

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^\lambda}. \quad (2.8)$$

### 2.6.3 Distancia de ciudad (City block distance)

Conocido por su nombre en inglés “City block Distance”, otros nombres son distancia de Manhattan, distancia de valor absoluto, la distancia mide la diferencia absoluta entre las coordenadas de dos objetos, siendo esta distancia un caso especial de la distancia de Minkowski con un  $\lambda = 1$  [PLS04] [Tek06], la fórmula es dada como:

$$d_{ij} = \sum_{k=1}^n |x_{ik} - x_{jk}|. \quad (2.9)$$

### 2.6.4 Distancia de Chebyshev

La distancia de Chebyshev o Tchebyschev, otro nombre es la distancia de máximo valor, se aplica para variables ordinales y cuantitativas, la distancia mide el máximo de la diferencia entre características de un par de objetos, siendo esta distancia un caso especial de la distancia de Minkowski con un  $\lambda = \infty$  [PLS04] [Tek06], la fórmula es dada como:

$$d_{ij} = \max_k |x_{ik} - x_{jk}|. \quad (2.10)$$

### 2.6.5 Distancia de Canberra

La distancia de Canberra mide la suma de la diferencia fraccional absoluta entre características de un par de objetos. Cada término fraccional tiene un rango entre 0 y 1. Si las dos características tienen un valor de 0, entonces se asume  $\frac{0}{0} = 0$ . Esta distancia es muy sensible a los pequeños cambios cuando las coordenadas son cercanas a cero [DD09] [LW67] [Tek06], la fórmula es dada como:

$$d_{ij} = \sum_{k=1}^n \frac{|x_{ik} - x_{jk}|}{|x_{ik}| + |x_{jk}|}. \quad (2.11)$$

### 2.6.6 Distancia de Bray-Curtis

La distancia de Bray-Curtis es algunas veces llamada distancia Sorensen. Esta distancia es usada comúnmente en botánica, ecología y ciencias ambientales. Cuando todas las coordenadas son positivas la función de la medida de distancia toma un valor entre cero y uno. Una distancia de cero corresponde exactamente a coordenadas similares. La función de la medida de distancia es “no definida” si los dos objetos tienen coordenadas con un valor de cero. La función de la medida de distancia es calculada usando la diferencia absoluta sobre la suma [LW67] [Tek06], la fórmula es dada como:

$$d_{ij} = \frac{\sum_{k=1}^n |x_{ik} - x_{jk}|}{\sum_{k=1}^n (x_{ik} + x_{jk})}. \quad (2.12)$$

### 2.6.7 Separación Angular

La Separación Angular a veces se denomina coeficiente de correlación, esta medida de distancia mide el coseno del ángulo entre dos vectores, este es una medida de similitud el cual toma valores en el rango -1 a +1, a diferencia de ser una medida de distancia. Cuando dos vectores son similares, la separación angular tomara un valor mayor. Si los dos vectores tienen un ángulo de 0, entonces se asume  $\frac{0}{0} = 0$  [DD09] [LW67], la Separación Angular entre dos vectores es definida como:

$$d_{ij} = \frac{\sum_{k=1}^n (x_{ik} \times x_{jk})}{\sqrt{\left(\sum_{k=1}^n x_{ik}^2 \times \sum_{k=1}^n x_{jk}^2\right)}}. \quad (2.13)$$

### 2.6.8 Coeficiente de Correlación

El Coeficiente de Correlación es otra medida de similitud a diferencia de ser una medida de distancia, a veces se denomina Coeficiente de Correlación Lineal o Coeficiente de Correlación Person. Este coeficiente es un caso espacial de la medida de similitud de Separación Angular tomando valores en un rango entre -1 y +1. La separación Angular se estandariza por medio de la centralización de las coordenadas en su media [DD09] [LW67], se define como:

$$d_{ij} = \frac{\sum_{k=1}^n (x_{ik} - \bar{x}_{ik})(x_{jk} - \bar{x}_{jk})}{\sqrt{\left(\sum_{k=1}^n (x_{ik} - \bar{x}_{ik})^2 \sum_{k=1}^n (x_{jk} - \bar{x}_{jk})^2\right)}}. \quad (2.14)$$

### 2.6.9 Distancia de Mahalanobis

La medida de distancia Mahalanobis algunas veces se denomina Distancia Cuadrática. La medida de distancia determina la diferencia o distancia entre dos grupos de objetos. Si dos grupos tienen su media  $\bar{x}_i$  y  $\bar{x}_j$ , estos dos objetos deben de tener el mismo número de características, pero podrían tener distinto número de muestras, la distancia de Mahalanobis es definida como [DD09] [LW67] [Tek06]:

$$d_{ij} = \left( (\bar{x}_i - \bar{x}_j)^T \times S^{-1} \times (\bar{x}_i - \bar{x}_j) \right)^{\frac{1}{2}}. \quad (2.15)$$

Donde: S es la matriz de covariancia y T es el número de muestras.

## 2.7 Medidas de Similitud Semánticas

Similitud semántica es un área muy atractiva por muchos años en el área de Inteligencia Artificial, Psicología y ciencia cognoscitiva. En recientes años el uso de WordNet [Fel98] y el uso de medidas de similitud a traído buenos resultados y muchas medidas de similitud se han propuesto.

Word Net es un producto resultante de la investigación realizada por la Universidad de Princeton – B10—Esta es la base de datos léxica bastante grande del lenguaje Ingles. En WordNet los verbos, adverbios, sustantivos y adjetivos son organizados utilizando una serie de relaciones semánticas, en total 28 tipos de relaciones, sinónimos, antónimos, hyponimos. Formando una estructura jerárquica y a su vez un grafo el cual es muy útil para el procesamiento del lenguaje natural.

Comúnmente se argumenta que semántica del lenguaje es generalmente capturada en los sustantivos o los sustantivos parafrásicales. Es por esto que la mayoría de las investigaciones se enfocan mayormente en medidas de similitud semántica entre sustantivos. WorNet incluye cuatro relaciones semánticas Hyponym/hypernym (is-a), part meronym/part holonym (part-of), member meronym/member holonym(member-of) y substance meronym/substance holonym (substance-of). Generalmente las relaciones derivadas de hyponym e hypernym son consideradas como relaciones de similitud entre dos conceptos.

Las medidas de similitud semánticas pueden ser usadas para tareas como la desambiguación, segmentación de texto, verificación de la consistencia y coherencia de ontologías. Las distintas medidas de similitud semánticas se pueden agrupar en cuatro clases, medidas basadas en el largo del camino (path), medidas basadas en el contenido de la información, medidas basadas en características y medidas hibridas.

WorNet desde un punto de vista es una estructura jerárquica, un grafo o una ontología. En el Procesamiento del Lenguaje Natural se construyen muchas estructuras jerárquicas, grafos y ontologías léxicas y semánticas con diferentes técnicas, algunos de estas estructuras incluyen miles de millones de relaciones [Varios corpus], nótese que las relaciones en WordNet han sido creadas por seres humanos. Las siguientes medidas de similitud son aplicables no solo a WordNet sino a otros modelos léxico y semánticos.

## 2.8 Medidas basadas en el largo del camino

### 2.8.1 Camino más corto

Esta medida solo toma en consideración el largo del camino entre dos palabras  $length(\text{palabra}_1, \text{palabra}_2)$ , esta considera que la similitud  $similitud(\text{palabra}_1, \text{palabra}_2)$  depende cuan cerca

se encuentran dos palabras en una taxonomía. De hecho, esta medida es una variación del método de cálculo de distancia [RMBB89] [BKA02] [VVR<sup>+</sup>05]. El cual es basado en dos observaciones, que el comportamiento del concepto se refleja en medida y la distancia conceptual entre dos palabras es proporcional al número de palabras que las separan, se define como:

$$\begin{aligned} similitud_{camino}(palabra_1, palabra_2) = \\ 2 \times profundidad\_maxima \times largo(palabra_1, palabra_2). \end{aligned} \quad (2.16)$$

### 2.8.2 Medida de Wu & Palmer

Esta medida de similitud incorpora un concepto más específico entre las dos palabras, para esto se mide la distancia entre la *palabra<sub>1</sub>* y el concepto más específico en la taxonomía, de igual forma para *palabra<sub>2</sub>*, *lso(palabra<sub>1</sub>, palabra<sub>2</sub>)*. Esta medida asume que la similitud entre dos conceptos es la función de longitud de trayectoria y la profundidad de medida basada en la trayectoria [WP94], se define como:

$$\begin{aligned} similitud_{WP}(palabra_1, palabra_2) = \\ \frac{2 \times profundidad(lso(palabra_1, palabra_2))}{longitud(palabra_1, palabra_2) + 2 \times profundidad(lso(palabra_1, palabra_2))}. \end{aligned} \quad (2.17)$$

Dónde: *lso(palabra<sub>1</sub>, palabra<sub>2</sub>)* es la distancia teniendo en cuenta la palabra más específica de los dos conceptos.

### 2.8.3 Medida de Leacock & Chodorow

Esta medida de similitud cuenta el número de palabras entre las dos palabras a medir, pero que solo tengan la relación “is-a” en la taxonomía. El valor se escala por la profundidad máxima de la relación “is-a” en la taxonomía. Un valor de relación se obtiene tomando el logaritmo negativo de este valor escalado [LC98], se define como:

$$similitud_{LC}(palabra_1, palabra_2) = -\log \left( \frac{longitud(palabra_1, palabra_2)}{2 \times profundidad\_maxima} \right) \quad (2.18)$$

### 2.8.4 Medida de Li

La medida de similitud de Li se deriva de manera intuitiva y empírica, se basa en el presupuesto que las fuentes de información son infinitas hasta cierto punto, mientras que los seres humanos comparamos la similitud de palabras utilizando un intervalo finito que va de algo complejo a algo no similar. La transformación de un modelo infinito a un modelo finito es no lineal. El modelo de medida combina la longitud más corta y la profundidad del concepto en una función no lineal [LBM03] [VVR<sup>+</sup>05], se define como:

$$\text{similitud}_{Li}(\text{palabra}_1, \text{palabra}_2) = e^{-\alpha * \text{longitud}(p_1, p_2)} \frac{e^{\beta * \text{profundidad}(\text{lso}(p_1, p_2))} - e^{-\beta * \text{profundidad}(\text{lso}(p_1, p_2))}}{e^{\beta * \text{profundidad}(\text{lso}(p_1, p_2))} + e^{-\beta * \text{profundidad}(\text{lso}(p_1, p_2))}} \quad (2.19)$$

Dónde:  $p_1, p_2$  son  $\text{palabra}_1, \text{palabra}_2$  respectivamente.

$\text{lso}(p_1, p_2)$  es la distancia teniendo en cuenta la palabra más específica de los dos conceptos. La medida monotónicamente incrementa con profundidad ( $\text{lso}(p_1, p_2)$ ) y decrece con profundidad longitud ( $p_1, p_2$ ).

Los parámetros  $\beta$  y  $\alpha$  deben de ser adaptados manualmente para mejorar la performance según [Res]  $\beta = 0.2$  y  $\alpha = 0.6$ .

## 2.9 Medidas basadas en el contenido de la información

Se asume que cada glosa de cada concepto provee información adicional a las relaciones entre los conceptos. Entre mayor la cantidad de elementos comunes en las glosas de dos conceptos mayor es su similitud.

### 2.9.1 Medida de Resnik

La medida de similitud de Resnik es basada en el contenido. Se asume que para dos conceptos, su similitud depende de la información contenida en la taxonomía y la que la subsume, considerar algo como parte de un conjunto más amplio o como caso particular sometido en la taxonomía [Res], se define como:

$$\text{similitud}_{Resnik}(\text{palabra}_1, \text{palabra}_2) = -\log p(\text{lso}(p_1, p_2)) = IC(\text{lso}(p_1, p_2)). \quad (2.20)$$

Dónde:  $p$  es la función de probabilidad IC es la información de los términos contenida en la taxonomía

### 2.9.2 Medida de Lin

La medida de similitud de Lin utiliza la cantidad de información necesaria para establecer la concordancia entre las dos palabras y la información necesaria para describir los términos [Lin98], se define como:

$$\text{similitud}_{\text{Lin}}(\text{palabra}_1, \text{palabra}_2) = \frac{2 \times \text{IC}(\text{lso}(p_1, p_2))}{\text{IC}(p_1) + \text{IC}(p_2)}. \quad (2.21)$$

### 2.9.3 Medidad de Jiang

Jiang argumenta que se puede obtener una distancia semántica, la cual es muy diferente a distancia, la distancia semántica se obtiene como la diferencia de las medidas de similitud de las palabras [JC97], se define como:

$$\text{dis}_{\text{Jiang}}(\text{palabra}_1, \text{palabra}_2) = (\text{IC}(p_1) + \text{IC}(p_2)) - (2 \times \text{IC}(\text{lso}(p_1, p_2))). \quad (2.22)$$

Funciones de soporte

$$\text{IC}(\text{palabra}) = -\log p(\text{palabra}). \quad (2.23)$$

$$p(\text{palabra}) = \frac{\text{frecuencia}(\text{palabra})}{N}. \quad (2.24)$$

$$\text{frecuencia}(\text{palabra}) = \sum_{p \in P(\text{palabra})} \text{contar}(p). \quad (2.25)$$

$$\text{IC}(\text{palabra}) = \frac{\log(\text{profundidad}(\text{palabra}))}{\log \text{profundidad maxima}}. \quad (2.26)$$

## 2.10 Medidas basadas en características

Esta medida es independiente de la taxonomía y subsumido de las palabras, por el contrario, intenta explotar las propiedades de una ontología para calcular la medida de similitud. Esta similitud asume que cada concepto esta descrito por una serie de palabras indicando las propiedades o características, siendo esta su definición o glosa en un diccionario o WordNet.

### 2.10.1 Similitud Asimétrica de Tversky

Tversky argumenta que la similitud no es simétrica por el contrario es asimétrica, las diferencias entre características de una clase inferior y una clase superior contribuyen a la evaluación de la similitud. Entre mayor es el número de características de la clase superior presentes en la clase inferior y menor es el número de características de la clase superior no presentes en la clase inferior, la similitud será mayor de la clase superior hacia la clase inferior y viceversa. Una consideración que se tiene que tener en cuenta en el modelo de Tversky es la utilización de dos valores  $\alpha, \beta$  los cuales deben ser propuestos por un experto, en este caso un ser humano [DHC09], se define como:

$$\begin{aligned} sim_{Tversky}(palabra_1, palabra_2) = \\ \frac{|palabra_1 \cap palabra_2|}{|palabra_1 \cap palabra_2| + \alpha |palabra_1 - palabra_2| + \beta |palabra_2 - palabra_1|}. \end{aligned} \quad (2.27)$$

$$\begin{aligned} sim_{Tversky}(palabra_2, palabra_1) = \\ \frac{|palabra_1 \cap palabra_2|}{|palabra_1 \cap palabra_2| + \alpha |palabra_2 - palabra_1| + \beta |palabra_1 - palabra_2|}. \end{aligned} \quad (2.28)$$

Donde:  $\alpha, \beta \geq 0$



# **Capítulo 3**

## **Métodos de Agrupamiento No Supervisado**

---

### **3.1 Introducción**

Reconocimiento de patrones puede verse de dos formas, aprendizaje de la variabilidad de las propiedades de las muestras las cuales pueden caracterizar una clase en particular, y decidir la clasificación de una nueva muestra, la membresía a una clase se puede dar tan solo por propiedades comunes entre los miembros. La clasificación puede ser supervisado y no supervisado, esto dependerá de la disponibilidad de etiquetas que identifican un patrón.

La clasificación no supervisada (en inglés clustering) es una técnica donde un número de patrones, usualmente vectores en un espacio multidimensional son agrupados en cúmulos (clusters) de tal forma que los patrones son similares en el mismo cúmulo o cluster, similares desde algún sentido y los patrones en distintos cúmulos o clusters son disimilares. El agrupamiento no supervisado (Clustering) es una tarea muy difícil puesto que existen una gran variedad de medidas de similitud y conceptos de disimilitud [BP07] [JMF99].

Para particionar un conjunto de datos en distintos grupos o cúmulos, uno tiene que definir una medida de similitud o una medida de proximidad, para la identificación de patrones y membresías. Se observa que en general una de las características fundamentales de las formas de los objetos es simetría, la cual es importante para mejorar el reconocimiento de patrones. La simetría es comúnmente encontrada en la naturaleza y es importante utilizar esta propiedad para la agrupación no supervisada. Por otro lado, la asimetría es intrínseca a la naturaleza ya que ningún objeto tiene exactamente los mismos patrones y es importante

utilizar esta propiedad para la agrupación no supervisada [Att55] [BS07] [BS08] [CSL02] [SB13] [SB08] [SC01].

El problema de la agrupación no supervisada es la selección de adecuados parámetros de similitud, así como un modelo rápido que procese grandes cantidades de datos. Se debe contar con múltiples objetivos los cuales pueden ejecutarse en paralelo para alcanzar una óptima agrupación no supervisada. Siendo un trabajo muy complejo e intensivo pudiendo no alcanzar la solución, pero si una solución aproximada. El objetivo de la agrupación no supervisada es poder tener un algoritmo robusto, rápido de ejecución y que se aproxime a la solución ya sea a través del uso de una sola técnica o múltiples técnicas.

En el presente trabajo usaremos el lema "cluster" y sus variaciones gramaticales para referirse a "agrupación no supervisada".

## 3.2 Definición de Agrupamiento no Supervisado

Agrupamiento no Supervisado también conocido como Clustering o clasificación no supervisada. Es un problema importante en minería de datos y reconocimiento de patrones. El cual tiene un número grande de aplicativos en distintas áreas. El Agrupamiento no supervisado es el proceso por el cual, dado un conjunto con patrones o características no etiquetados, organizados en un espacio vectorial y/o matricial, son agrupados en grupos o cúmulos de tal forma que los patrones o características sean similares en algún sentido y las características entre distintos grupos o cúmulos sean disimiles en el mismo sentido. Matemáticamente, el agrupamiento no supervisado o clustering partitiona el espacio vectorial o matricial en K regiones basados en alguna métrica de similitud o métricas de disimilitud, donde K podría ser conocido a priori o no. El objetivo de los métodos de agrupamiento no supervisado o clustering es de transformar un conjunto de datos en una matriz de grupos, donde cada grupo represente un patrón en particular [JD88] [TG74].

### 3.2.1 Técnicas de Agrupamiento no Supervisado

Existe una gran cantidad de técnicas de agrupamiento no supervisado o clustering en la literatura, Las técnicas tradicionales se pueden clasificar en 4 tipos o clases: jerárquico, particional, basado en la densidad y basado en mallas. algunos ejemplos de las técnicas jerárquicas son "linkage clustering", "average linkage clustering", y "complete linkage clustering". K-means es una técnica de particionamiento. DBSCAN es una técnica basada en la densidad y STING es basado en mallas [TG74] [WYM<sup>+</sup>97] [JMF99] [JD88] [EKS<sup>+</sup>96].

### 3.3 Técnicas de particionamiento

Este método produce un número de grupos donde cada valor pertenece a un grupo. Un grupo es generalmente representado por punto central o centroide que es visto como el punto que describe a todos los puntos en este grupo en particular. La definición del punto central o centroide dependerá en los tipos de datos que se están agrupando. En el caso de valores reales, el centroide es definido como la media aritmética de todos los elementos en el grupo. En el caso de agrupamiento de documentos, un centroide puede ser una lista de aquellas palabras que ocurren en los documentos en el grupo. Si el agrupamiento genera grupos grandes, el grupo puede ser nuevamente particionado obteniendo nuevos centroides. Algunos de los algoritmos más populares en la literatura son K-means, K-medoids, fuzzy C-means clustering, y expectation maximization clustering [BP07].

#### 3.3.1 K-Means Clustering

El algoritmo de K-Means es una técnica iterativa de agrupamiento no supervisado, el cual transforma K datos en grupos de tal manera que en cada grupo el centro se reduce al mínimo.

$$J = \sum_{j=1}^n \sum_{k=1}^k u_{kj} \times \|\bar{x}_j - \bar{z}_k\|^2 \quad (3.1)$$

Donde:

$u_{kj}$  es igual a 1; si el  $j$ -simo punto pertenece a  $k$ ; y es igual a 0 caso contrario.

$\bar{z}_k$  denota el centro del grupo  $k$ ; y  $\bar{x}_j$  denota el punto  $j$ -simo de los datos.

K-Means inicialmente inicia una serie K de centroides de forma aleatoria, y la primera partición es formada usando el criterio mínimo de distancia. Subsecuentemente los centros de cada grupo son actualizados con la media del grupo. Una nueva partición se da después de cada actualización. Este proceso se repite hasta que se cumpla uno de los siguientes puntos:

1. El centro del grupo no cambia con las subsecuentes iteraciones.
2. El valor de  $j$  es menor de un límite predefinido.
3. El máximo número de iteraciones se ha agotado.

El objetivo es poder minimizar  $J$ .

$$\frac{dJ}{d\bar{z}_k} = 2 \sum_{j=1}^n u_{kj} (\bar{x}_j - \bar{z}_k) (-1) = 0, k = 1, 2, \dots, k \quad (3.2)$$

$$\sum_{j=1}^n u_{kj} \bar{x}_j - \bar{z}_k \sum_{j=1}^n u_{kj} = 0 \quad (3.3)$$

$$\bar{z}_k = \frac{\sum_{j=1}^n u_{kj} \bar{x}_j}{\sum_{j=1}^n u_{kj}} \quad (3.4)$$

Para un agrupamiento no supervisado no superpuesto o "crisp clustering",  $\sum_{j=1}^n u_{kj}$ , el objetivo es la pertenencia de un elemento a un solo grupo, quedando la ecuación de la siguiente forma:

$$\bar{z}_i^* = \frac{\sum_{\bar{x}_j \in C_i} \bar{x}_j}{n_i} \quad (3.5)$$

Para garantizar que  $j$  sea minimizado, la segunda derivada debería ser positiva. Note que la parte derecha de la ecuación es positiva lo cual indica que en cada interacion  $j$  tendrá el valor mínimo [ELL01] [JD88].

$$\frac{d^2 j}{d \bar{z}_k^2} = 2 \sum_{j=1}^n u_{kj} \quad (3.6)$$

El algoritmo de K-means tiene varias limitaciones, como se muestra en [SI84], K-means podría converger a un valor no optimo, esto dependerá de la selección de los centroides iniciales, el cual es dado de forma aleatoria. Así también una solución para un problema de grandes datos (big data), no podría ser alcanzado en un periodo de tiempo razonable [Spä80]. Más aún este algoritmo asume que los grupos son de forma hiperesférica y más o menos de igual tamaño. K-means no es robusto para elementos aislados (outliers).

### 3.3.2 K-Medoid Clustering

El algoritmo de agrupamiento no supervisado K-medoid es entre otros uno de los más usados [TK06]. Este es una extension del algoritmo K-means, donde el concepto de medoid es usado en vez de la media. Medoid tiene un punto representativo de un conjunto de datos, cuyo promedio de disimilitud a todos los puntos de datos es mínimo. Se puede considerar como el punto más centrado en el grupo.

Al igual que en K-means, K-medoid también trata de minimizar el error cuadrático total de todo el conjunto de datos, la distancia entre los puntos marcados para estar en un grupo y un punto designado como el centro del grupo. La técnica de agrupación K-medoid minimiza una suma de diferencias por pares en lugar de una suma de las distancias euclidianas al cuadrado, es más robusto al ruido y los valores atípicos en comparación con las técnicas de

K-means. La técnica de agrupación K-medoid agrupa los datos en  $K$  grupos de los cuales  $K$  es conocido [TK06] [BS12].

Los pasos son:

1. Número de puntos de  $K$  son seleccionados al azar a partir del conjunto de  $n$  puntos.
2. Iniciar cuando los medoids  $K$ .
3. Cada punto de datos se asigna a la medoid más cercana (aquí la medida de distancia depende de las necesidades del usuario y el tipo de datos).
4. Calcular el error cuadrático total para la nueva partición.
5. Repetir los pasos 2,3 y 4 hasta que no exista ningún cambio en las asignaciones o el cambio llegue a un umbral.

### 3.3.3 Fuzzy C-Means Clustering

El algoritmo Fuzzy C-means es muy usado, el cual usa el principio de fuzzy set para mejorar la partición de un conjunto dado [Bez13]. Las particiones difusas se representan como una matriz asociando cada fila a uno de los  $c$  grupos y cada columna a uno de los elementos de  $X$ , de forma tal que el valor en la fila  $i$  y la columna  $j$  indique la pertenencia del elemento  $j$  al grupo  $i$ . Más formalmente, el conjunto de las particiones difusas se puede definir como:

$$M_{fcn} = \left\{ U \in R^{c \times n} \mid \sum_{i=1}^c u_{ik} = 1, \sum_{k=1}^n u_{ik} > 0, \forall i \text{ y } u_{ik} \in [0, 1]; 1 \leq i \leq c; 1 \leq k \leq n \right\}. \quad (3.7)$$

Los algoritmos Fuzzy c-Means son algunos de los principales algoritmos utilizados en el agrupamiento difuso y pertenecen a una clase de algoritmos basados en funciones objetivo. Estos algoritmos definen un criterio de agrupamiento en la forma de una función objetivo que depende de la partición difusa. El procedimiento, en sentido general, consiste en minimizar iterativamente esta función hasta obtener una partición difusa óptima.

$$J\mu(U, Z) = \sum_{i=1}^c \sum_{k=1}^n (u_{ik})^\mu D^2(\bar{z}_i, \bar{x}_k) \quad (3.8)$$

$U \in M_{fcn}$  es una partición de matriz fuzzy; donde:  $\mu \in [1, \infty]$  es la ponderada exponencial de cada miembro fuzzy;  $Z = [\bar{z}_1, \dots, \bar{z}_c]$  representa el centro de un grupo  $c$ ;  $\bar{z}_i \in R^N$ ; y  $D(\bar{z}_i, \bar{x}_k)$  es la distancia de  $\bar{x}_k$  desde un  $i$ -simo centro de grupo [TK06]. Una estrategia común para

una aproximación a la solución de minimizar la ecuación 3.8 es dada a través de la técnica iterativa de Picard [Bez13].

La agrupación resultante de la técnica de Fuzzy generalmente es fina, pero este tiene el mismo problema que K-means, ambos se quedan atascados en óptimo local, dependiendo del número de grupos a generar, el cual es un parámetro a priori [BMM07].

### 3.4 Agrupación basado en la distribución

Este tipo de agrupamiento no supervisado asume que las características de los grupos siguen una distribución particular. Uno de los más prominentes algoritmos es la expectativa de maximización (*EM*, expectation maximization). Este es un enfoque iterativo que se enfoca en determinar los parámetros de la distribución de la probabilidad que tienen maximum likelihood. El objetivo de este algoritmo es de dividir los  $n$  puntos en  $K$  grupos, de tal forma que el total de las probabilidades de ocurrencia de los grupos es maximizada. Los parámetros del algoritmo son: el conjunto de puntos  $n$ , el número de grupos  $K$ , el error de convergencia aceptado  $E$ , y el máximo número de iteraciones  $i$ . Dos pasos son ejecutados en cada iteración [DLR77]:

1. Expectativa: Cálculo de probabilidad que cada punto pertenezca a cada grupo.

$$P(C_j | x) = \frac{|\Sigma_j(t)|^{\frac{1}{2}} \exp^{n_j} P_j(t)}{\sum_{k=1}^M |\Sigma_k(t)|^{\frac{1}{2}} \exp^{n_k} P_k(t)}. \quad (3.9)$$

2. Maximización: Volver a estimar las probabilidades de distribución de cada grupo después de la asignación.

$$\mu_j(t+1) = \frac{\sum_{k=1}^N P(C_j | x_k) x_k}{\sum_{k=1}^N P(C_j | x_k)}. \quad (3.10)$$

El algoritmo termina después de cumplir una serie de  $i$  iteraciones o cuando el parámetro de convergencia converja. La probabilidad e ocurrencia de cada grupo es calculado con la media de probabilidades ( $C_j$ ):

$$P_j(t+1) = \frac{1}{N} \sum_{k=1}^N P(C_j | x_k). \quad (3.11)$$

Esta técnica asume que los grupos obtenidos siguen una distribución normal. Si los grupos no siguen esta distribución normal, el algoritmo de *EM* no realizará la partición de forma apropiada.

### 3.5 Las técnicas de agrupamiento jerárquico

El agrupamiento jerárquico es un método de análisis de grupos el cual busca construir una jerarquía de grupos, pueden ser de dos tipos:

1. **Aglomerativas:** Este es un acercamiento ascendente, cada observación comienza en su propio grupo, y los pares de grupos son mezclados mientras uno sube en la jerarquía.
2. **Divisivas:** Este es un acercamiento descendente, todas las observaciones comienzan en un grupo, y se realizan divisiones mientras uno baja en la jerarquía.

Estas fusiones y fraccionamientos generan algunos nuevos grupos, los cuales se determinan de una manera voraz. Los resultados de la agrupación jerárquica se presentan habitualmente en un dendrograma [TK06].

#### Linkage Clustering

La técnica de Linkage es un algoritmo no-iterativo basado en un criterio de conectividad local. El algoritmo no usa un solo punto por el contrario usa un conjunto de relaciones  $r_{jk}$  entre un par de objetos. El valor de la relación  $r_{jk}$  representa el grado en que el objeto  $j$  y  $k$  están relacionados, según una relación binaria  $\rho$  z.

El algoritmo inicia considerando cada punto como un grupo, y **Single Linkage** calcula la distancia entre dos grupos  $C_i$  y  $C_j$  como:

$$\delta_{SL}(C_i, C_j) = \min_{\bar{x} \in C_i, \bar{y} \in C_j} \{d(\bar{x}, \bar{y})\}. \quad (3.12)$$

Dónde:  $d(\bar{x}, \bar{y})$  es algún tipo de distancia entre objetos  $\bar{x}$  y  $\bar{y}$ , basado en la distancia se pueden juntar dos grupos, donde su distancia es mínima, sucesivamente la distancia del nuevo grupo a los otros grupos es nuevamente calculado, estos pasos se repiten hasta que el número deseado de grupos  $K$  es alcanzado. Lo positivo de este algoritmo es que es independiente de la forma de los grupos y trabaja con atributos numéricos y categóricos; las desventajas son su cálculo computacional y la inutilidad de generar grupos superpuestos.

En algoritmo de **Average Linkage** la distancia entre dos grupos  $C_i$  y  $C_j$  es definida como:

$$\delta_{AL}(C_i, C_j) = \frac{\sum_{\bar{x} \in C_i, \bar{y} \in C_j} \{d(\bar{x}, \bar{y})\}}{|A| \times |B|}. \quad (3.13)$$

En el caso del algoritmo **Complete Linkage** la distancia entre dos grupos  $C_i$  y  $C_j$  es definida como:

$$\delta_{CL}(C_i, C_j) = \max_{\bar{x} \in C_i, \bar{y} \in C_j} \{d(\bar{x}, \bar{y})\}. \quad (3.14)$$

Tanto Average Linkage como Complete Linkage tienen los mismos pasos que el algoritmo Single Linkage. Complete Linkage identifica grupos compactos de tamaño similar, pero es muy sensitivo a valores atípicos. Por otro lado Single Linkage genera grupos largos encadenados [JMF99] [TK06].

## 3.6 Agrupamiento basado en densidad

El algoritmo de Agrupamiento basado en densidad es una técnica que utiliza un criterio local para la agrupación. Grupos consisten de regiones en el espacio de datos, donde puntos forman un área densa los cuales están separados por puntos con una densidad baja, considerado como ruido. Los grupos formados pueden tomar distintas formas y sus miembros pueden estar distribuidos dentro de la región de forma aleatoria, es así que este tipo de técnica puede identificar un grupo de cualquier forma, pero el algoritmo depende de la densidad de los puntos. Existen varios algoritmos enfocados en agrupamiento basado en densidad.

**DBSCAN** (Density-Based Spatial Clustering of Applications with Noise), algoritmo más popular, el cual se enfoca en identificar grupos bien definidos, el concepto que se considera es el de "densamente alcanzable". El algoritmo está basado en conectar puntos dentro de cierto umbral de distancia. Aun así, sólo conecta aquellos puntos que satisfacen un criterio de densidad. Otra propiedad interesante de DBSCAN es que su complejidad es bastante baja  $O(n \log n)$ , el cual requiere un número lineal de consultas de rango en la base de datos en cada corrida, por tanto no hay ninguna necesidad de correrlo varias veces. Cabe mencionar que la asignación de membresía no es simétrica. El algoritmo debe satisfacer los dos criterios siguientes:

1. Todos los puntos dentro de un grupo particular, deben ser mutuamente conectados densamente.
2. Si un punto está conectado densamente a cualquier punto de la agrupación, entonces el punto es incluido en el grupo.

DBSCAN requiere dos parámetros  $\epsilon$  la distancia requerida para calcular la alcanzabilidad de densidad y el mínimo de puntos requeridos para formar un grupo. El algoritmo selecciona de forma aleatoria un punto con el que va a trabajar, con este punto se calcula la densidad con sus vecinos y de acuerdo a la distancia requerida  $\epsilon$ , se forma un grupo, se verifica el mínimo número de miembros para formar un grupo, si cumple esta condición el grupo con el punto y todos los vecinos, el grupo es formado, de lo contrario el punto es marcado como un punto atípico. Posteriormente este punto podría ser considerado en la formación de algún otro grupo y ser parte de este nuevo grupo en un ambiente con una distancia requerida  $\epsilon'$ . Si un punto viene a formar parte de un grupo, todos sus vecinos que cumplan la distancia requerida  $\epsilon$  también formaran parte del grupo. El proceso continúa seleccionando puntos aleatorios hasta que no existan puntos no asignados a un grupo [EKS<sup>+</sup>96] [EKS<sup>+</sup>96].

**GDBSCAN** (Generalized Density-Based Spatial Clustering of Applications with Noise), es una generalización de los mismos autores a los algoritmos basados en vecindad y densidad, DBSCAN. Los parámetros distancia requerida  $\epsilon$  y mínimo número de miembros en un grupo se eliminan del algoritmo original y se trasladan a los predicados. Por ejemplo, en los datos del polígono, la vecindad podría ser cualquier polígono de intersección, mientras que la densidad utiliza las áreas de polígonos en lugar de sólo el contador de objeto [EKS<sup>+</sup>96] [SEKX98].

**OPTICS** (Ordering points to identify the clustering structure), este algoritmo es una técnica basada en la agrupación jerárquica de densidad. Recordemos que DBSCAN detecta grupos de densidad definida por el usuario. OPTICS aborda una de las principales debilidades de DBSCAN, el problema de la detección de grupos significativos en los datos de densidad variable. OPTICS trabaja de la siguiente manera, los puntos de la base de datos se ordenan de forma lineal, de tal manera que los puntos que son espacialmente más cerca se convierten en vecinos. Además, una distancia especial se almacena para cada punto que representa la densidad que necesita ser aceptado para un grupo con el fin de obtener que los dos puntos pertenezcan al mismo grupo. OPTICA es capaz de producir una estructura jerárquica basada en la densidad del conjunto de datos dado. El grafo generado mediante el uso de este algoritmo se llama la traza de accesibilidad que muestra los grupos de diferentes densidades, así como agrupaciones jerárquicas [ABKS99].

### 3.7 Agrupamiento basado en mallas

Los algoritmos de agrupamiento basados en mallas son populares con grandes cantidades de datos en un espacio multidimensional. La complejidad es similar a muchos algoritmos de agrupación el cual depende linealmente del tamaño de los datos, manejar gran cantidad de

datos es uno de sus puntos fuertes, es similar a los algoritmos basados en densidad, pero no toma en cuenta la densidad de los vecinos sino sus alrededores, puntos que se encuentran al rededor. En términos generales existen 5 pasos [GMW07] [GB02]:

1. La generación de la estructura de malla, esto se puede hacer mediante la partición del espacio de datos en un número finito de mallas.
2. La densidad de cada celda se calcula basándose en el número total de puntos dentro de la malla.
3. Las mallas son ordenadas según sus densidades.
4. Se calculan los centros de cada grupo.
5. Las mallas vecinas se recorren.

**STING** (statistical information grid-based clustering method), este algoritmo es prácticamente utilizado para agrupar datos espaciales, el algoritmo divide el espacio en mallas y las mallas son representadas por una estructura jerárquica, la raíz de la jerarquía es considerada como nivel 1, el cual está constituida por sus mallas hijos, una celda en nivel  $i$  contiene la unión de sus hijos de nivel  $i + 1$ , cada celda tiene cuatro hijos, representando un cuadrante del padre. Información estadística relativa a los atributos de cada celda de la malla, como la media, máximo y valores mínimos son pre calculados y almacenados en cada celda.

STING ofrece varias ventajas, el cálculo basado en mallas es de consulta independiente, debido a que la información estadística almacenada en cada celda representa la información de resumen de los datos de la celda de la cuadrícula, independientes de la consulta; otra ventaja es la estructura de malla facilita el procesamiento en paralelo y actualización incremental; y otra ventaja es la eficiencia del algoritmo el cual es una gran ventaja. STING pasa a través de los datos una vez para calcular los parámetros estadísticos de las células, y por lo tanto la complejidad de tiempo de agrupaciones de generación es  $O(n)$ , donde  $n$  es el número total de datos. Después de generar la estructura jerárquica, el tiempo de procesamiento de consultas es  $O(g)$ , donde  $g$  es el número total de celdas de la cuadrícula en el nivel más bajo, que es por lo general mucho menor que  $n$ . STING es sólo aplicable a datos bidimensionales [WYM<sup>+</sup>97] [WYM99].

### 3.8 Mecanismo de división y combinación

Los algoritmos de división y combinación, de forma general trabajan de la siguiente manera; dado un conjunto de datos, los datos que están ubicados en una región local son parte del mismo grupo. Esta es la forma como estos algoritmos trabajan, sus pasos son:

1. El conjunto es dividido en subgrupos de tal forma que el subconjunto sea homogéneo como sea posible.
2. Repetidamente combinar los subconjuntos vecinos con cierto criterio de combinación hasta que el número especificado de grupos es obtenido.

Los algoritmos basados en este mecanismo descomponen un problema complejo en problemas simples, y son efectivos en conjuntos de datos complejos.

Chameleon [KHK99] y CSM (Cohesion Based Self-Merging) [DK82] son dos algoritmos basados en la división y combinación. Chameleon construye un grafo basado en k-nearest-neighbor de los datos, y utiliza hMetis [KK98] para particionar el grafo, el cual minimiza el corte de las aristas del grafo. Este define una similitud de los grupos, el cual consiste de dos factores, interconectividad relativa y relativa proximidad, y se combina los pares de los subgrupos más similares, esto se realiza de forma repetitiva.

CSM utiliza K-means para dividir el conjunto de datos en una serie de subgrupos, define la cohesión de los grupos para medir la similitud entre grupos y combina los subgrupos. Ambos métodos pueden hacer frente a los problemas de agrupamiento complejo.

### 3.9 Medidas de validación de agrupación

La evaluación del desempeño de la agrupación efectuada por los distintos algoritmos de agrupamiento es un área de investigación. A continuación, presentaremos las medidas más comúnmente utilizadas como criterios de evaluación del desempeño para algoritmos de agrupamiento. Una variedad de medidas se ha propuesto en la literatura.

La función más simple para medir la calidad de una partición es "cut" y la forma más directa para construir la partición es resolver el problema mincut. Dado el número  $k$  de particiones, el enfoque mincut elige una partición  $C_1, \dots, C_k$  que reduce al mínimo.

$$cut(C_1, C_2, \dots, C_k) = \frac{1}{2} \sum_{i=1}^k W(C_i, \bar{C}_i). \quad (3.15)$$

Para  $k = 2$  mincut es relativamente simple y un problema fácil de resolver. Pero en la práctica, las particiones no son a menudo satisfactorias. Más específicamente, la solución tiende a separar un vértice individuo del resto de la gráfica que no se espera. En la partición, cada grupo debe ser lo suficientemente grande. Existen dos populares métodos para incluir este aspecto de una partición, RatioCut y normalized cut.

Cut Ratio se define como, dado un grupo de vértices  $C$  es la suma de los pesos de las aristas de conexión de  $C$  hacia el resto del grafo, normalizado por el tamaño de  $C$ .

$$RatioCut(C_1, \dots, C_k) = \frac{1}{2} \sum_{i=1}^k \frac{W(C_i, \bar{C}_i)}{|C_i|}. \quad (3.16)$$

Probablemente la función de calidad más popular para la partición de grafos es de normalized cut. El normalized cut de un grupo de vértices  $C$  es la suma de los pesos de las aristas de conexión  $C$  al resto del grafo, normalizado por peso total de las aristas en  $C$  y el peso total de las aristas del resto de los vértices en el grafo.

$$NCut(C_1, \dots, C_k) = \frac{1}{2} \sum_{i=1}^k \frac{W(C_i, \bar{C}_i)}{d(C_i)}. \quad (3.17)$$

Donde  $d(C_i)$  es grado total del grupo  $C_i$ .

El enfoque de RatioCut y Normalized Cut es que las particiones sean balanceadas. Los grupos con baja Normalized Cut representan buenas comunidades porque están bien conectados entre sí y escasamente conectados con el resto del grafo [Kin82] [HdCC04].

La conductancia es una medida relacionada y se define como:

$$Conductancia(C) = \frac{\sum_{i \in C, j \in \bar{C}} W(i, j)}{\min(\sum_{i \in C} d(i), \sum_{i \in \bar{C}} d(i))}. \quad (3.18)$$

El Normalized Cut o conductancia de una partición de un grafo en  $k$  grupos  $C_1, \dots, C_k$  es la suma del Normalized Cut o conductancia de las particiones individuales  $C_i$  para  $i = 1, \dots, k$ .

Otra medida popular de la eficiencia de partición de un grafo es la modularidad. Esta medida es la suma de las diferencias de cada grupo, entre la fracción de las aristas internas y la fracción de aristas en el interior del grupo y se define como:

$$Q = \sum_{i=1}^k \left[ \frac{W(C_i, C_i)}{e} - \left( \frac{d(C_i)}{2e} \right)^2 \right]. \quad (3.19)$$

Donde  $C_i$ s son grupos,  $e$  es el número de aristas, y  $d(C_i)$  representa el grado total de grupo  $C_i$  [N125].

Por desgracia, la optimización de cualquiera de estas funciones es NP-hard.

Por otro lado la evaluación de grafos de redes basado en Ground-Truth o conocimiento previo de los sub grupos que deben de ser obtenidos. Jaewon Yang y Jure Leskovec [BB98] Identificaron una serie de métodos para la evaluación de grafos de redes basados en Ground-Truth, el cual identifica cuatro clases, basado en la conectividad interna (Internal Density, Edges Inside, Average Degree, Fraction Over Median Degree, Triangle Participation Ratio), basados en la conectividad externa (Expansion, Cut Ratio), combinación de la conectividad interna y externa (Conductance, Normalized Out Degree Fraction, Average Out Degree

Fraction, Flake Out Degree Fraction), y basados en el modelo de la red (Modularity). [BB98] argumenta que todas las técnicas están íntimamente relacionadas y en particular para la separabilidad la técnica de Conductance es la que tiene mejores resultados, mientras que para densidad, cohesividad y agrupamiento la técnica Triangle Participation Ratio es la más adecuada.

### 3.10 Agrupación no Supervisada Basado en Grafos

Recordando el problema básico de Agrupación no Supervisada, donde  $X = x_1, \dots, x_n$  es un conjunto de puntos,  $S = (S_{ij})_{i,j=1,\dots,n}$  es la matriz de similitud en el que cada elemento indica su similitud entre dos puntos  $x_i$  y  $x_j$ ; donde la similitud es  $S_{ij} \geq 0$ . Una forma de representar estos datos es a través de la construcción de un grafo, en el cual cada vértice un punto del conjunto de datos, y el peso de las aristas representan la similitud entre dos puntos o vértices.

El problema de Agrupación no Supervisada con respecto a un grafo se formulada como la partición del grafo en subgrafos, de tal manera que la cohesión interna es mayor que la cohesión de los vértices y aristas hacia otro subgrafo.

Un grafo es un triple  $G = (V, E, W)$  donde  $V = v_1, \dots, v_n$  es el conjunto de vértices,  $E \subseteq V \times V$  es el conjunto de aristas, y  $W = (w_{ij})_{i,j=1,\dots,n}$  es llamada una matriz adyacente en el cual cada elemento representa un peso no negativo  $(w_{ij}) \geq 0$  entre dos vértices  $w_i$  y  $w_j$ .

La actual investigación esta enfocada a un agrupamiento duro donde los subgrafos no se sobreponen, los siguientes algoritmos son relacionados a este tipo de agrupación.

Modelado de transformar un problema en un grafo, se define el significado del vértice y aristas en el grafo, se calcula los pesos de las aristas para un grafo ponderado, Luxburg [VL07] declaró tres métodos más comunes para la construcción de un grafo:  *$\varepsilon$ -neighborhood graph, k-nearest neighbor graph, and fully connected graph*.

Se pueden categorizar los algoritmos de agrupamiento en dos clases: división y aglomeración, a su vez cada uno tiene sub-clases. Los algoritmos de división tienen un mecanismo top-down y recursivamente dividen el grafo en sub grafos. En contraposición a los algoritmos aglomerantes tienen un mecanismo bottom-up e iterativamente unen conjuntos unitarios de vértices en sub grafos. estos dos algoritmos también son llamados jerárquicos por que producen grupos a diferentes niveles; un grupo es refinado por sus subgrafos y es más genérico en su grafo padre 3.1 [CJ10].

Los algoritmos cut-based se basan en el teorema max-flow y min-cut (Ford y Fulkerson, 1956), el cual establece que "el valor del flujo máximo es igual al costo de la reducción mínima de corte". El algoritmo de Kernighan-Lin (Kernighan y Lin, 1970) divide el grafo en dos de forma recursiva, con el objetivo de minimizar la densidad inter-grupos. Flake

et al. (2003) propone un algoritmo de corte basado en heurísticas el cual optimiza los resultados, este algoritmo se clasifica como un algoritmo de agrupación supervisado puesto que parametriza el corte del grafo según algunas heurísticas, según el caso del grafo.

Los algoritmos basados en el teorema espectral de grafos y la matriz de Laplacian ( $L$ ). La conexión entre la agrupación y espectro de la matriz Laplaciana, se basa en la siguiente proposición, La multiplicidad  $K$  de eigenvalue 0 de  $L$  es igual al número de componentes conectados en el grafo. Luxburg (2006) argumenta tres formas de las matrices de Laplacian, una no-normalizada el cual busca optimizar la taza de corte ratiocut, y dos normalizadas la cual busca optimizar el número de cortes  $ncut$ .

El éxito de la agrupación espectral se basa principalmente en el hecho de que no hace suposiciones fuertes en la forma de los grupos y puede resolver problemas muy generales como espirales entrelazadas, que el algoritmo agrupación k-means no logra realizar la agrupación. Por desgracia, la agrupación espectral podría ser inestable bajo diferentes opciones de grafos y parámetros, más aún en un modelo de grandes datos llega no práctico, todo está basado en el cálculo de los eigenvectores de matriz de Laplacian el cual es  $O(|V|^3)$ , sin embargo, cuando la matriz tiene pocos datos la complejidad se reduce a  $O(|V||E|)$  aplicando eficientemente el algoritmo de Lanczos. No obstante, la rápida generación de datos los cuales requieren que sean agrupados, los algoritmos espectrales tendrían que procesar todos los datos cuando un nuevo dato es incluido en el conjunto de datos, otro enfoque es requerido como los algoritmos de agrupación por flujo, como el presentado en esta investigación PAC [STC<sup>+</sup>15c].

Los algoritmos basados en el paradigma de partición multiniveles de grafos (Karypis y Kumar, 1999) el cual consiste en tres pasos, una partición inicial y dos de refinación. Un primer enfoque realiza una partición en dos sub grupos en multiniveles de forma recursiva; un segundo enfoque realiza la partición en  $k$  grupos de forma genérica y posteriormente realiza un afinamiento. El segundo enfoque tiene una menor complejidad en comparación a la primera, más aún en algunos grafos el desempeño es mejor.

Los algoritmos basados en random walk es muy probable que visite varios vértices en un sub grupo antes de moverse al siguiente sub grupo. Uno de los mejores enfoques es el de Markov clustering algorithm (MCL) (Dogen, 2000). MCL iterativamente hace uso de dos operaciones, expansión e inflación, hasta que el grafo converja. La operación de expansión simula random walk y la operación de inflación modela el descenso inter-sub grupos. El objetivo es reducir inter-subgrupos y aumentar intra-sub grupos, su complejidad es  $O(m^2|V|)$  donde  $m$  es el número de recursos utilizados en cada vértice.

Los algoritmos basados en la hipótesis, los enlaces entre subgrupos se encuentran en el camino mas corto. Girvan y Newman (2003) propusieron el concepto de *betweeness* el

cual es el número de caminos más cortos conectando cualquier par de vértices que pasa a través de sus aristas. El algoritmo de forma iterativa retira la arista con mayor *betweenness*, el algoritmo tiene una complejidad de  $O(|C||V|^2)$ . Fortunato et al. (2004) utiliza *information centrality* un remplazo de *betweenness*, consiguiendo mejores resultados pero con un costo de complejidad mayor  $O(|C||V|^3)$ .

Los algoritmos de aglomeración como el propuesto por Newman (2004) el cual inicia cada vértice como un subgrupo unitario y de forma iterativa une dos sub grupo unitario, según el resultante de la modularidad sea un incremento sustancial o un decremento mínimo. el algoritmo converge cuando sólo queda un sólo grupo, luego se selecciona los sub grupos con mayor modularidad, la complejidad es de  $O(|V||E|)$ .

Los algoritmos basados en División y Aglomeración como el propuesto en este trabajo de investigación PAC el cual describiremos en detalle en el capítulo 5.

### 3.11 Medidas de Calidad y Evaluación para Grafos

Una medida es una función objetivo que evalúa la calidad de una agrupación, así llamado medición de calidad. Mediante la optimización de la medición de la calidad, podemos obtener la agrupación "óptima". Vale la pena señalar que la medida de calidad no se debe confundir con la medida de similitud de vértices. Por otra parte, hay que distinguir medida de calidad de la medida de evaluación.

La diferencia principal es que la medida de calidad de una agrupación identifica directamente una agrupación que cumple una propiedad deseable, mientras que la medida de evaluación valora la calidad de un agrupamiento mediante la comparación de los grupos generados con un resultado pre conocido o ground-truth.

En adición a las medidas presentadas en el punto 3.9, existen algunas medidas utilizadas en particular para grafos tales como medidas de densidad (intra-grupos, inter-grupos), medidas basadas en cut-based (ratio cut, ncut, performace, expansion, conductance, bicriteria), así como la modularidad. La optimización de cada una de las medidas es NP-hard, muchos algoritmos que tratan de resolver el problema de optimización con una complejidad polinómial, tienen resultados de agrupación sub óptimos.

La medida de calidad no se traduce necesariamente en la efectividad en aplicaciones reales con respecto al agrupación pre conocida o *ground – truth*. Por lo tanto, una medida de evaluación desempeña el papel de evaluar la efectividad del agrupamiento si coincide con la agrupación pre conocida o *ground-truth*.

Zheng Chen et al [CJ10] plantean dos preguntas: ¿Qué restricciones (propiedades, criterios) debe satisfacer una medida de evaluación ideal? y ¿Las medidas de evaluación

Tabla 3.1 Algoritmos de agrupamiento en grafos.

Categoría	Algoritmo	Complejidad
<b>División</b>		
cut-based	Algoritmo Kernighan-Lin (Kernighan and Lin, 1970)	$O( V ^3)$
cut-based	Algoritmo cut-clustering (Flake et al., 2003)	$O( V )$
spectral	unnormalized spectral clustering (Luxburg, 2006)	$O( V  E )$
spectral	normalized spectral clustering I (Luxburg, 2006; Shi and Malik, 2000)	$O( V  E )$
spectral	normalized spectral clustering II (Luxburg, 2006; Ng, 2002)	$O( V  E )$
spectral	iterative conductance cutting (ICC) (Kannan et al., 2000)	$O( V  E )$
spectral	geometric MST clustering (GMC) (Brandes et al., 2007)	$O( V  E )$
spectral	modularity oriented (White and Smyth, 2005)	$O( V  E )$
multilevel	multilevel recursive bisection (Karypis and Kumar, 1999)	$O( V logK)$
multilevel	multilevel K-way partitioning (Karypis and Kumar, 1999)	$O( V  = KlogK)$
random	Markov Clustering Algorithm (MCL) (Dongen, 2000)	$O(m^2 V )$
shortest path	betweenness (Girvan and Newman, 2003)	$O( V  E ^2)$
shortest path	information centrality (Fortunato et al., 2004)	$O( V  E ^3)$
<b>Aglomeración</b>		
agglomerative	modularity oriented (Newman, 2004)	$O( V  E )$
<b>División y Aglomeración</b>		
Paradigmatic modularity	PAC (2015)	$O(V * AGV(degree)^2)$
<b>Donde:</b>		
$ V $ es el número de vértices		
$ E $ es el número de aristas		
$K$ es el número de grupos		
$m$ es el número de recursos asignados a cada vértice		

Tabla 3.2 Medida de Evaluación Zheng Chen et al [CJ10]

Categoría	Medida de Evaluación
set mapping	purity, inverse purity, F-measure
pair counting	rand index, Jaccard Coefficient, Folks and Mallows FM
entropy	entropy, mutual information, VI, V
editing distance	editing distance
coreference resolution	MUC (Vilain et al.,1995), B-Cubed (Bagga and Baldwin, 1998), CEAf (Luo, 2005)

propuesto que restricciones deben de satisfacer?, así Zheng Chen et al argumenta. Para la primera pregunta Dom (2001) desarrolla una técnica parametrizada para describir la calidad de una agrupación y propone cinco propiedades, Meila (2003) argumenta 12 propiedades relacionadas con las medidas de entropía, Amigo et al. (2008) propuso cuatro restricciones *homogeneity, completeness, ragbag, and clustersizevs.quantity*. Estas cuatro propiedades tienen una ventaja sobre las propuestas de Dom (2001) y Meila (2003), por una razón, estas cuatro propiedades pueden describir las más importantes propiedades propuestas por Dom (2001) y Meila (2003). Para responder la segunda pregunta Zheng Chen et al [CJ10] concluye:

1. Todas las medidas excepto B-Cubed fallan la restricción rag bag y única medida B-Cubed puede satisfacer todas las cuatro restricciones.
2. Las dos medidas basadas en entropía (VI y V) y MUC sólo fallan a la restricción de rag bag.
3. Todas las medidas en la categoría set mapping fallan restricción de completeness.
4. Todas las medidas en la categoría pair counting fallan el agrupamiento de size vs. quantity.
5. CEAf, por desgracia, falla en las restricciones de homogeneity, completeness, rag bag.

Estándares de benchmarks para algoritmos de agrupamiento basados en grafos de redes, como el de Girvan y Newman entre otros, no dan cuenta de las características importantes de los grafos de redes reales, como la fat-tailed distribuciones del grado de nodos y tamaño de la comunidad. Por lo tanto, Santos Fortunato [Abn07] propuso una novedosa clase de benchmark para grafos de redes, en la que las distribuciones del grado de los nodos y el tamaño de la comunidad son exponenciales con un exponente parametrizado. La técnica

propuesta por Santos Fortunato utiliza una variación de *normalized mutual information NMI*, conocido el Ground-Truth del grafo se utiliza NMI para comparar los subgrupos generados por el algoritmo con los grupos que deberían haberse generado, basado esto en el conocimiento previo de los subgrupos, Ground-Truth. El autor denota que la comparación es mucho mejor a diferencia de calcular la conectividad o densidad o la conductividad, puesto que conocido el Ground-Truth del grafo, es mucho más directa la evaluación, ver cuán preciso es el algoritmo.

Es esta última técnica que usaremos para realizar nuestras evaluaciones en distintos casos, así también nos valdremos de los utilitarios proveídas por el Dr. Santos Fortunato.

# Capítulo 4

## Agrupación Paradigmática

---

### 4.1 Introducción

Las medidas de similitud son esencialmente para resolver muchos problemas de reconocimiento de patrones como la clasificación, agrupamiento y problemas de extracción de datos. Muchas medidas de similitud son categorizadas como relacionadas, sintácticas y semánticas. En esta tesis presentamos la incorporación de una novedosa medida de similitud, Coeficiente Unilateral Jaccard (uJaccard), el cual no sólo toma en consideración el espacio entre dos puntos sino la semántica entre ellos.

¿Cómo podemos extraer información significativa de un grafo grande y escaso? El abordaje tradicional es enfocarse en técnicas genéricas de agrupamiento, descubriendo cúmulos densos en redes de grafos, no obstante, se omite patrones interesantes tales como las relaciones paradigmáticas.

En esta tesis proponemos un novedoso algoritmo de agrupamiento no supervisado para grafos, modelando las relaciones de los nodos y utilizando el análisis paradigmático. Nosotros explotamos las relaciones de los nodos para extraer el conjunto de significantes. Los grupos encontrados representan una vista diferente de los grafos, el cual proporciona información detallada de la estructura de grafos y grafos escasos. Nuestro algoritmo propuesto PaC, definido por las siglas en inglés “Paradigmatic Clustering” para la agrupación de grafos realiza un análisis paradigmático el cual es soportado por una medida de similitud asimétrica uJaccard. En contraste a los métodos de agrupación tradicionales, nuestro algoritmo produce resultados meritorios en las tareas de desambiguación lingüística de palabras. Se han realizado experimentos extensivos y se ha realizado un análisis empírico para evaluar nuestro algoritmo en datos reales y sintéticos.

Desde Euclides hasta hoy, muchas medidas de similitud se han propuesto para considerar diferentes escenarios en distintas áreas. Particularmente en la última centuria, las medidas de similitud son usadas para comparar diferentes tipos de datos, lo cual es fundamentalmente importante para la clasificación de patrones, agrupamiento, y problemas de extracción de información [DHS].

Relaciones de similitud generalmente han sido dominadas por modelos geométricos en los cuales son representados por puntos en un modelo de espacio de Euclides [She74]. La similitud es definida como “tener las mismas o casi las mismas características” [Fel05b], mientras que las medidas de distancia son definidas como “La propiedad creada por el espacio entre dos objetos o puntos”. Todas las funciones de distancia deben de satisfacer tres axiomas básicos: minimalidad y la igualdad de auto-similitud, la simetría y la desigualdad del triángulo.

$$d(i, i) = d(j, j) \leq d(i, j) \quad (4.1)$$

$$d(i, j) = d(j, i) \quad (4.2)$$

$$d(i, j) + d(j, k) \geq d(i, k) \quad (4.3)$$

Para los objetos  $i$ ,  $j$  y  $k$ , donde  $d(i, j)$  es la distancia entre objeto  $i$  y  $j$ . Bridge [Bri98] argumenta que existe evidencia empírica de la violación de los tres axiomas. Más aún, existen modelos geométricos de similitud que toman la asimetría en consideración [Nos91]. Nosofsky señala que una serie de modelos bien conocidos para los datos de proximidad asimétricos están estrechamente relacionados con la similitud aditiva y modelo de sesgo [Hol79]. Tversky [Tve77b] ha propuesto un modelo diferente con el fin de superar el supuesto de métrica de modelos geométricos. Uno de los puntos fuertes de los modelos de contraste es su capacidad para explicar los juicios de similitud asimétricos. La asimetría de Tversky puede a menudo ser caracterizada en términos de sesgo de estímulo y se determina por la prominencia relativa de los estímulos.

$$\text{sim}(a, b) = \frac{|A \cap B|}{|A \cap B| + \alpha |A - B| + \beta |B - A|}, \quad (4.4)$$

$$\alpha, \beta \geq 0$$

$A$  y  $B$  presentan un conjunto de características para los objetos  $a$  y  $b$ , respectivamente. El término en el numerador es una función del conjunto de características compartidas, Una medida de similitud, y los dos últimos términos en el denominador mide la disimilitud: ( $\alpha$ ) y ( $\beta$ ) son números reales ponderados; donde ( $\alpha$ )! = ( $\beta$ ). [JBG12], [WW05] y [LPCM99] también propone una medida de similitud asimétrica basada en el trabajo de Tversky. Pero

todas las propuestas incluyen un sesgo de estímulo, Juicio de similitud asimétrica, donde Tversky se refiere a un juicio de un ser humano. Hoy en día las medidas de similitud están incorporadas en muchos de los algoritmos para clasificación, agrupación y otras tareas. Todas estas técnicas están dejando de lado la semántica de cada vértice y sus relaciones con otros vértices y aristas.

En un grafo dirigido, la similitud entre  $U$  y  $Z$  no es igual a la distancia de  $Z$  y  $U$ , esto es debido a las características intrínsecas de un grafo dirigido. Las similitudes son diferentes porque los canales son distintos. De acuerdo con la teoría de información de Shannon nosotros podríamos argumentar que cada vértice es fuente de energía con una entropía media que es compartida entre sus canales, y mientras esa información fluye entre los canales del vértice, nosotros tenemos que considerarlo en la similitud. Una similitud no es para todos los casos o tareas.

En el procesamiento del lenguaje natural, donde la similitud entre dos palabras no es simétrica  $\text{sim}(\text{word}_a, \text{word}_b) \neq \text{sim}(\text{word}_b, \text{word}_a)$ . WordNet [Fel05b] presenta 28 diferentes tipos de relaciones. Esas relaciones tienen dirección y no son simétricas, ellos ni siquiera son sinónimos porque cada palabra sinónimo tiene una semántica particular, significado y uso, pero son similares. Por lo tanto si dos palabras tienen similitudes o distancia simétrica, estas dos palabras serían la misma. La característica Paradigmática es intrínseca al lenguaje, le deja al hablante intercambiar palabras con otras, palabras semánticamente similares [Sah06]. En esta tesis nosotros nos enfocamos en un análisis paradigmático para proveer soporte a nuestra propuesta de similitud Coeficiente de Similitud Unilateral Jaccard (uJaccard).

#### 4.1.1 Agrupamiento

Datos de diferentes dominios, como de redes sociales, bio-informática, redes y procesamiento del lenguaje natural pueden ser modelados como grafos. Cada grafo contiene comunidades o grupos que representan el mundo real, cada grupo es caracterizado por una densidad de los enlaces internos que de los enlaces externos; el problema es definir qué es un grupo y cómo identificarlo. [FC12] ha identificado tres diferentes categorías que definen comunidades o grupos dentro de un grafo: definición Local, definición Global y similitud de vértices. La definición local es cuando las comunidades son agrupadas basadas en sus vértices y sus vecinos inmediatos, sin considerar el resto del grafo. La definición global es cuando las comunidades son agrupadas basadas en un análisis de las características de la comunidad y con respecto a los otros nodos y/o comunidades en el grafo. Por último, definición de la similitud de vértices es cuando una comunidad es agrupada basada en la similitud de sus vértices, cuando un criterio cuantitativo es seleccionado para evaluar la similitud entre cada par de vértices. El problema en este caso es identificar una medida adecuada de similitud.

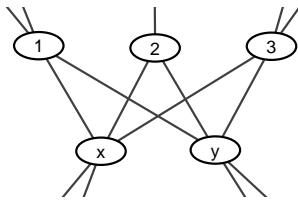


Figura 4.1 Equivalencia Estructural.

[YL15] ha identificado 13 medidas de agrupación las cuales están basadas en la conectividad interna y externa de un grupo o comunidad en un grafo, así como la conectividad de interna y externa de los grupos o comunidades y el modelo de redes. Todos los algoritmos analizan enlaces internos y externos en diferentes formas y miden el grado de correlación de todas los grupos o comunidades encontradas.

## 4.2 Definición de Similitud Paradigmática

### 4.2.1 Conceptos Básicos de las Estructuras Paradigmáticas

El análisis paradigmático es un proceso que identifica entidades que no son directamente relacionadas, pero son relacionadas por sus propiedades, relación entre entidades y su capacidad de intercambio [DSB11]. En el lenguaje de la razón por la que se tienden a utilizar formas morfológicamente no relacionadas en oposición comparativa es hacer hincapié en la semántica, esto se logra por medio de la sustitución y transposición de palabras con una significante similar. La similitud no es definida por las reglas sintácticas, pero sino más bien por su uso en el lenguaje. En algunos casos este uso no gramaticalmente o sintácticamente correcto, pero es de común uso. Nosotros definimos el significante como el grado de relación entre entidades del mismo grupo, donde no todos los miembros del grupo tienen el mismo grado de relación, esto es por el hecho de que un miembro de un grupo podría pertenecer a más de un grupo.

### 4.2.2 Extended Similitud Paradigmática

Dos vértices en un grafo son equivalentes estructuralmente si comparten muchos de los mismos vecinos de la red. Equivalencia regular es muy sutil, dos vértices no necesariamente comparten los mismos vecinos, pero ellos tienen vecinos quienes que son similares entre ellos, los vecinos pueden ser de profundidad  $n$  [LW71] [WR83]. Nosotros usaremos la equivalencia regular como las bases de uJaccard.

### Coeficiente de Similitud Unilateral Jaccard

Para calcular la similitud paradigmática nosotros iniciamos con una pregunta, ¿es el coeficiente de similitud entre los vértices  $V_a$  a  $V_c$  el mismo que el coeficiente de similitud entre los vértices  $V_c$  a  $V_a$ ? Si argumentamos que el coeficiente de similitud es el mismo, estamos argumentando que vértices  $V_a$  y  $V_c$  son el mismo y es claro que no es el caso, los dos vértices tienen diferentes aristas.

Un problema con la similitud de Tversky [Tve77b] es que los estimados para  $\alpha$  y  $\beta$  son estímulos de tendencia, generalmente ser un factor proveído por un ser humano. Otras medidas de similitud que están basadas en Tversky, tiene el mismo problema. Por otro lado en esta tesis se propone el uso de un medida que no incluye esta tendencia, se propone una versión modificada del coeficiente de Jaccard(4.5)(4.6), Coeficiente de Similitud Unilateral Jaccard(4.7)(4.8), utilizado para identificar el coeficiente de similitud entre los vértices  $V_a$  a  $V_c$  y entre los vértices  $V_c$  a  $V_a$ , donde  $uJaccard(V_a, V_c) \neq uJaccard(V_c, V_a)$ .

$$Jaccard(a, c) = \frac{|a \cap c|}{|a \cup c|} \quad (4.5)$$

$$Jaccard(c, a) = \frac{|c \cap a|}{|c \cup a|} \quad (4.6)$$

$$uJaccard(a, c) = \frac{|a \cap c|}{|degree(a)|} \quad (4.7)$$

$$uJaccard(c, a) = \frac{|c \cap a|}{|degree(c)|} \quad (4.8)$$

Donde  $a$  y  $c$  son vértices en un grafo de red.  $(a \cap c)$  es la intersección de aristas entre vértices  $a$  y  $b$ . El  $degree(a)$  es el grado del vértice  $a$  y  $degree(c)$  es el grado del vértice  $c$ .

Si  $uJaccard$  se acerca a 0, esto significa que no son similares en lo absoluto. El objetivo de usar  $uJaccard$  es identificar cuan similar es el vértice  $a$  de  $c$  y viceversa.  $uJaccard$  puede calcularse entre dos vértices conectados,  $uJaccard$  también puede ser calculado entre vértices que no están calculados directamente, pero que están conectados por un vértice intermedio. El número de vértices intermedios pueden ser de 1 a  $n$ , no recomendamos comparaciones muy profundas, puesto que la semántica entre los vértices comienza a perder significado. Por lo tanto, se sugiere  $max(n) = 3$  para el caso de procesamiento del lenguaje natural. Para las calculaciones no consideramos el número de vértices intermedios puesto que nos enfocamos en el flujo y no en la transformación que sucede en cada vértice intermedio.

Siguiendo la teoría de información de Shannon, podríamos argumentar que cada vértice es fuente de energía con una entropía media que es compartida entre sus canales. En

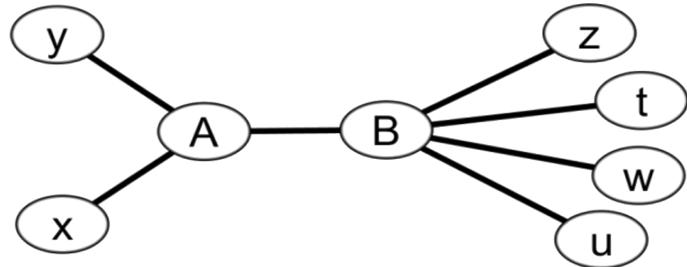


Figura 4.2 Simple graph

figura 4.2 se muestra un grafo pequeño, la similitud entre vértice  $A$  y  $B$  es de la siguiente forma  $uJaccard(A,B) = 1/3 = 0.333$ ;  $uJaccard(B,A) = 1/5 = 0.2$  y  $Jaccard(A,B) = 1/7 = 0.143$ ;  $Jaccard(B,A) = 1/7 = 0.143$ . El coeficiente de Jaccard provee una medida general de similitud simétrico, entre vértices  $A - B = B - A$ , sin embargo  $uJaccard$  provee un coeficiente de similitud asimétrico  $A - B \neq B - A$ .

### 4.3 Evaluacion Experimental de $uJaccard$

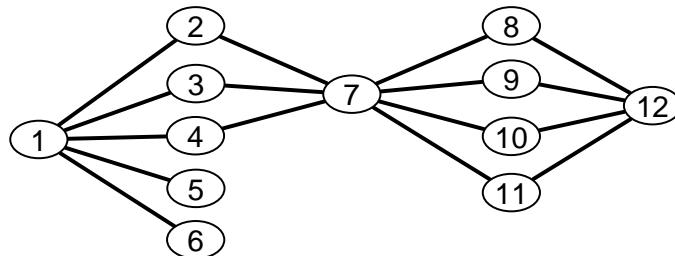


Figura 4.3 Toy graph.

#### 4.3.1 Pruebas Pequeñas

Usando el Coeficiente de Similitud Unilateral Jaccard (4.7)(4.8) podemos utilizar un enfoque paradigmático, para agrupar vértices, Figura 4.3 muestra un grafo pequeño con 12 vértices y 16 aristas, Siguiendo el análisis paradigmático, se ha determinado que vértice 12 y 7 pertenece a grupo  $P$  porque ellos tienen el mismo número de aristas conectadas al mismo conjunto de vértices. Vértice 1 también pertenece al grupo  $P$  porque tiene 3 de sus 5 aristas conectadas al mismo conjunto de vértices que 12 y 7. El grado es menor al de 12 y 7, por tanto, el coeficiente  $uJaccard$  con mayor similitud del vértice 1 es el vértice 7. De la misma manera podemos determinar que vértices 8, 9, 10, y 11 pertenecen al grupo  $Q$  porque ellos tienen el mismo número de aristas conectadas al mismo conjunto de vértices. De la misma

forma vértice 2, 3 y 4 pertenecen al grupo  $R$ , y vértice 5 y 6 pertenecen al grupo  $O$ . En este ejemplo se puede observar fácilmente el enfoque paradigmático, donde uno o más vértices pertenecen al mismo grupo si ellos tienen los mismos o similares vecinos, pero los vecinos pertenecen a otro grupo.

Tabla 4.1 Calculos de uJaccard de la figura 4.3

uJaccard (V1,V7) = 3/5 = 0.600
uJaccard (V7,V1) = 3/7 = 0.428
uJaccard (V7,V12) = 4/7 = 0.571
uJaccard (V12,V7) = 4/4 = 1.000
Jaccard (V1,V7) = 3/9 = 0.333
Jaccard (V7,V1) = 3/9 = 0.333
Jaccard (V7,V12) = 4/7 = 0.571
Jaccard (V12,V7) = 4/7 = 0.571

Siguiendo el Coeficiente de Similitud Unilateral Jaccard y el enfoque paradigmático, los resultados del grafo en figura 4.3 se muestran en la Tabla 4.1. Se puede notar que la uJaccard provee mejor información que Jaccard, esto es debido a que uJaccard considera la noción de similitud unilateral. Tabla 4.2 muestra tres ejemplos de pequeños grafos, en los cuales presentamos una comparación entre Jaccards y uJaccard. Como se muestra uJaccard mejora el coeficiente de Jaccard.

Table 4.2 muestra tres grafos pequeños, en el cual presentamos una comparación entre Jaccard y uJaccards. Como se muestra uJaccard provee un coeficiente más rico que el de Jaccard, mejorando la calidad de similitud.

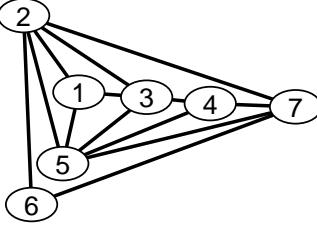
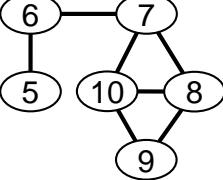
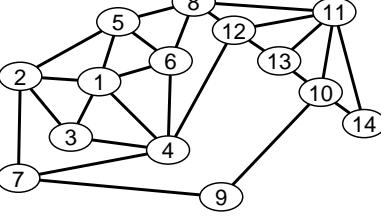
### 4.3.2 Cortando Grafos

En teoría de grafos, cortar es particionar los vértices de un grafo en dos subconjuntos desconectados. Existen muchas técnicas y algoritmos para cortar un grafo, pero en muchos casos los grafos son muy difíciles de cortar, debido a su distribución simétrica de vértices.

Se muestra en figura 4.4 que el nodo 1 podría pertenecer a los subconjuntos 2, 3, 4 o al subconjunto 5, 6; para resolver este problema nosotros usaremos Coeficiente de Similitud Unilateral Jaccard para encontrar el subconjunto más similar al nodo 1. Tabla 4.3 muestra las similitudes del nodo 1 a otros nodos a un nivel de profundidad, el cual es igual para todos los casos. Consecuentemente si utilizamos dos niveles de profundidad encontramos que  $uJaccard(1,3)$  tiene una similitud fuerte sobre el resto de nodos. Podemos concluir que el nodo 1 pertenece al subconjunto 2, 3, 4.

En la Figura 4.4 también el nodo 1 podría pertenecer al grupo 2, 3, 4 o al grupo 5, 6, 7 o al grupo 8, 9, 10, 11; es así que usamos uJaccard, pudiendo resolver este problema. Tabla

Tabla 4.2 pruebas de uJaccard en pequeños grafos

	<b>uJaccard</b> $\text{sim}(2,5)=4/4$ $\text{sim}(5,2)=4/6$  <b>Jaccard</b> $\text{sim}(2,5)=4/6$ $\text{sim}(5,2)=4/6$
	<b>uJaccard</b> $\text{sim}(7,5)=1/3$ $\text{sim}(5,7)=1/1$  <b>Jaccard</b> $\text{sim}(7,5)=1/3$ $\text{sim}(5,7)=1/3$
	<b>uJaccard</b> $\text{sim}(2,4)=3/4$ $\text{sim}(4,2)=3/5$  <b>Jaccard</b> $\text{sim}(2,4)=3/6$ $\text{sim}(4,2)=3/6$

4.4 muestra los resultados de similitud para el nodo 1 a otros nodos en la red en diferentes niveles. Grupo 8, 9, 10, 11 presenta las más fuerte similitud, es así que podemos concluir que nodo 1 pertenece al grupo 8, 9, 10, 11.

### 4.3.3 Redes Sociales

Nosotros hemos probado uJaccard con dos redes sociales, los cuales se representan como grafos, el primero es la rede de coautoría de científicos [New06], la segunda es la rede de actores de Hollywood (The Internet Movie Database: <ftp://ftp.fu-berlin.de/pub/misc/movies/database/>).

La primera red es una red de coautoría de científicos trabajando en teoría de redes, compilada por [New06]. Queremos encontrar a quien es similar científico Newman, bajo un enfoque paradigmático. Como se muestra en tablas 4.5 Newman tiene una similitud paradigmática con 3 científicos que son Callaway, Strogatz and Holme. Por otro lado, los científicos que son más similares paradigmáticamente a Newman son Adler, Aberg and

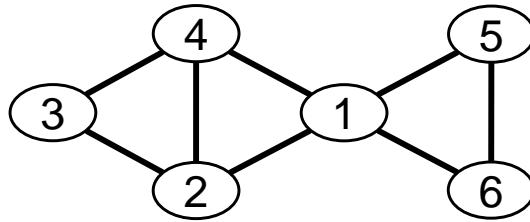


Figura 4.4 Grafo Pequeño.

Tabla 4.3 Cortando el grafo (figura 4.4) usando uJaccard

1 level deep		2 levels deep	
uJaccard(1,4)	1/4	uJaccard(1,2)	1/4
uJaccard(1,2)	1/4	uJaccard(1,3)	2/4
uJaccard(1,5)	1/4	uJaccard(1,4)	1/4
uJaccard(1,6)	1/4	uJaccard(1,5)	1/4
-	-	uJaccard(1,6)	1/4

Aharony. uJaccard se ha calculado en 2,4 y 4 niveles de profundidad desde y hacia Newman. Newman es más similar a Strogatz pero el científico que es más similar a Newman es Adler y Strogatz, más aun Strogatz mayor similitud es hacia Newman.

Para la segunda red, hemos creado la segunda red social de actores de Hollywood, nos basamos en “The Internet Movie Database”. Bajamos data de los actores y actrices, datos que incluían títulos de las películas en los que ellos habían trabajado, también bajamos datos de los mejores 1000 y 2500 actores y actrices. La red está compuesta de nodos que representan actores y actrices, las aristas representan las películas donde ellos han trabajado juntos. Un nodo es creado por cada persona, con su nombre como llave, cuando dos personas han trabajado en la misma película, una arista es creada entre los dos nodos. La primera rede representa los mejores 1000 actores y actrices quienes trabajaron en 41,719 películas con un total de 113,478 aristas. La segunda red representa los mejores 250 actores y actrices quienes trabajaron en 15,831 películas con un total de 14,096 aristas. Para esta prueba hemos removido aristas duplicadas.

El resultado de búsqueda en la rede de mejores 250 y 1000 actores y actrices, utilizando uJaccard con el enfoque paradigmático se presenta en Tabla 4.6 y Tabla 4.7. En la Tabla 4.6 nos enfocamos en *Tom Cruise*, encontramos que *Tom Cruise* es el más similar a *Julia Roberts* pero *Julia Roberts* es más similar a *John Travolta*, *Tom Cruise* es el tercero en la lista de *Julia Roberts*. Claramente no existe una similitud simétrica entre *Julia Roberts* y *Tom Cruise*. Más aun *Julia Roberts* no es la más similar hacia *Tom Cruise*, el más similar hacia *Tom Cruise* es *Heath Ledger*. Por lo tanto esto confirma que uJaccard ayuda en la identificación de similitud, particularmente con una similitud asimétrica. Tabla 4.6 muestra un escenario

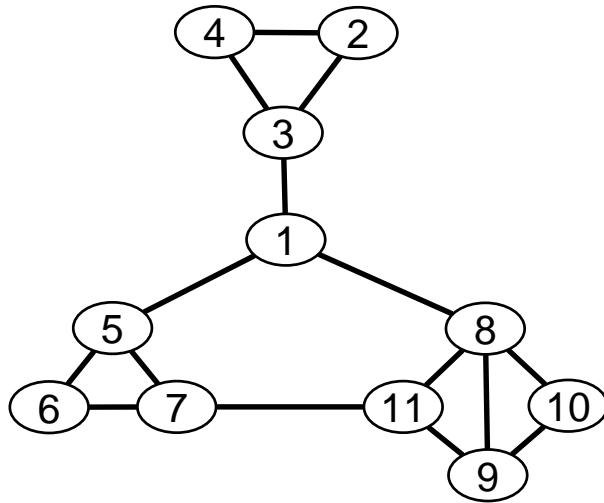


Figura 4.5 Grafo Pequeño.

Tabla 4.4 Cortando el grafo (figura 4.5) usando uJaccard

2 levels deep		3 levels deep	
uJaccard(1,4)	1/3	uJaccard(1,4)	1/3
uJaccard(1,2)	1/3	uJaccard(1,2)	1/3
uJaccard(1,6)	1/3	uJaccard(1,6)	1/3
uJaccard(1,7)	1/3	uJaccard(1,7)	2/3
uJaccard(1,9)	1/3	uJaccard(1,9)	2/3
uJaccard(1,11)	1/3	uJaccard(1,11)	2/3
uJaccard(1,10)	1/3	uJaccard(1,10)	1/3

similar entre *Tom Cruise*, *Tom Hanks* y *Joan Allen* en la rede de los 1000 mejores actores y actrices, esto confirma la usabilidad de uJaccard.

- Dado un *actor A*
- Se busca por *actores T* donde el *actor A* sea similar a los *actores T*
- De la lista de *actoresT* extraer los actores con mayor similitud en *actoresB*
- Se busca por *actores W* donde el *actorT<sub>n</sub>* sea similar a los *actores W*
- De la lista de *actoresW* extraer los actores con mayor similitud en *actoresA*
- Esto es echo para analisar si *actorA* y *actorB* son recíprocamente similares
- Esto se realizo en la red de los 250 mejores actores y la red de los mejores 1000 actores.

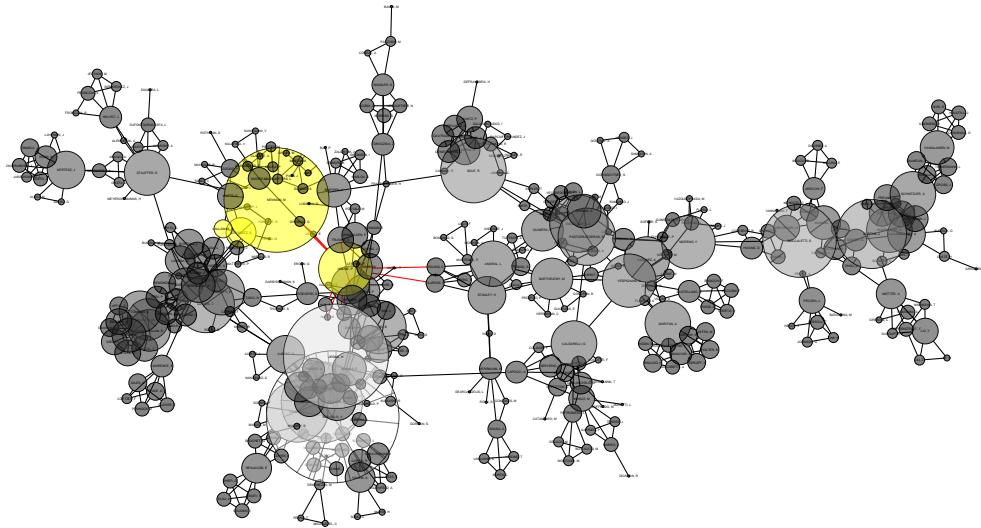


Figura 4.6 Coautoría rede de científicos.

El nodo seleccionado es el del científico Newman, Callaway, Strogatz, Holme.

En la Tabla 4.7 presentamos los resultados de los mejores 250 actores y actrices, en este caso no enfocamos en *Anthony Quinn* y *Jack Nicholson*. Nosotros comenzamos por buscar actores que *Anthony Quinn* es similar, luego buscamos actores que son lo más similar a *Anthony Quinn* en 1 y 2 niveles de profundidad. Podemos notar que *Anthony Quinn* es más similar a *Tom Hanks* pero el actor que es más similar a *Anthony Quinn* es *Antonio Banderas*, mientras *Anthony Quinn* está en la posición 153 de los más similares a *Tom Hanks*. *Antonio Banderas* es más similar a *Samuel Jackson* y no a *Anthony Quinn*, mientras que *Anthony Quinn* está en la posición 53 de los más similares a *Antonio Banderas*. Por consiguiente podemos concluir que *Anthony Quinn* y *Tom Hanks* no son similares simétricamente más bien son similares asimétricamente. Tabla 4.7 también muestra similar escenario para con *Jack Nicholson*.

#### 4.4 Agrupación de grafos usando la estructura paradigmática

En esta tesis nosotros proponemos una propiedad sub-estructural entre dos vértices cual quiera en un grafo, una propiedad paradigmática. En esta nueva propiedad vértices son equivalentes si ellos están conectados de la misma forma o similarmente, siendo esta una similitud de vértices de acuerdo con [FC12], así como una propiedad basada en una regularidad estructural como ya lo definimos anteriormente, soportada por el análisis paradigmático.

Tabla 4.5 Calculando uJaccard de la figure 4.5, búsqueda paradigmática.

Búsqueda paradigmática del científico Newman					
2 niveles de profundidad		3 niveles de profundidad		4 niveles de profundidad	
Científico Newman es similar a:					
Callaway	0.15	Strogatz	1.63	Strogatz	8.25
Strogatz	0.15	Holme	1.59	Callaway	7.85
Watts	0.15	Kleinberg	1.59	Watts	7.81
Hopcroft	0.11	Sole	1.59	Kleinberg	7.18
Científicos que son similares a Newman:					
Adler	0.33	Aberg	0.50	Aberg	2.50
Aharony	0.33	Adler	0.66	Adler	14.0
Aleksiejuk	0.50	Aharony	0.66	Aharony	14.0
Ancelmeyers	0.66	Alava	0.50	Alava	1.00
Araujo	0.33	Albert	0.10	Albert	0.50

Nosotros extendemos el concepto incorporando la similitud asimétrica propuesta en esta tesis Coeficiente de Similitud Unilateral Jaccard, uJaccard.

El análisis paradigmático busca identificar los diferentes paradigmas o conjunto de existentes significados los cuales manifiestan el contenido de los datos. Este aspecto de análisis envuelve el descubrir la existencia de una similitud paradigmática subyacente. El análisis paradigmático involucra la comparación y contrastar cada uno de los significantes que están presente en los datos. La prueba de la conmutación puede ser usada para identificar distintivos significado y para definir su significante, determinando si a cambio en el nivel del significado conlleva a un cambio en el nivel del significado [dS59]. El enfoque paradigmático es común en lingüística, redes sociales y otros [CCGV10].

PaC por sus siglas en inglés Paradigmatic Clustering, Agrupamiento Paradigmático, se enfoca en la similitud paradigmática de los vértices. Vértices son similares si ellos comparten o tienen el mismo número de aristas conectadas al mismo conjunto de vértices directa o indirectamente.

En esta tesis presentamos un algoritmo original de agrupamiento para grafos basado en un enfoque paradigmático, asistido por un grafo bipartito [FB04] y mejorado por el cálculo de modularidad. El algoritmo PaC es rápido y no es complejo, este tiene un tiempo de complejidad de  $O(n * \text{avg}(degree)^2)$ , este es adecuado para grandes grafos, altamente paralelizable, incremental y realiza agrupación por flujo. PaC es mejorado con el cálculo de modularidad para grafos el cual se realiza de arriba hacia abajo, la mejora se da después de la primera corrida de PaC.

Tabla 4.6 Similitud de uJaccard entre mejores 100 y 250 actores y actrices.

En busca de la similitud paradigmática entre los mejores 1000 y 250 actores y actrices.

Top 1000 actors, cruise tom is similar to:			
		roberts julia	
roberts julia	0.405	travolta john	0.418
hanks tom	0.401	hanks tom	0.412
jackson samuel	0.399	jackson samuel	0.407
douglas michael	0.397	cruise tom	0.399
eastwood clint	0.393	spacey kevin	0.399
Top 250 actors, cruise tom is similar to:			
		hanks tom	
hanks tom	0.430	douglas michael	0.425
douglas michael	0.420	cruise tom	0.421
eastwood clint	0.420	jackson samuel	0.418
spacey kevin	0.413	travolta john	0.414
jackson samuel	0.410	spacey kevin	0.411
Who is similar to cruise tom:			
top 1000 actors		top 250 actors	
ledger heath	0.490	allen joan	0.496
bacon kevin	0.488	balk fairuza	0.495
crowe russell	0.482	bello maria	0.488
gibson mel	0.482	collins pauline	0.487
benigni roberto	0.482	aiello danny	0.487

#### 4.4.1 Estructura Paradigmática

El análisis paradigmático es un proceso que identifica entidades las cuales no son relacionadas directamente, pero son relacionadas por sus propiedades, la relación entre entidades y si son intercambiables [dS59]. En lenguaje, la razón por que nosotros tratamos de usar forma morfológica no relacionada en comparaciones es para enfatizar la semántica el cual se realiza por substitución y transposición de palabras con un significante similar. La similitud no es definida por el conjunto sintáctico de reglas usadas en el lenguaje, en algunos casos su uso no es correcto gramatical o sintáctico, pero es de uso común.

PaC se basa en el principio de equivalencia regular, algunos vértices son paradigmáticamente intercambiables si ellos están apoyados en el mismo conjunto de aristas y sus sub-estructura son equivalentes a un cierto grado. Por ejemplo, en algunos contextos uno podría usar la palabra *family* para referirse a tus *friends*, las palabras no son relacionadas sintácticamente o semánticamente, pero en algunos contextos son comunes para intercambiar las palabras para ampliar el significado. Pero la similitud de la palabra *family* a la palabra

Tabla 4.7 Similitud de uJaccard de los mejores 250 actores y actrices en niveles.

en búsqueda de la similitud paradigmática entre los mejores 250 actores y actrices a 2 y 3 niveles de profundidad.

Top 250 actors, are similar to:			
quinn anthony		nicholson jack	
hanks tom	0.451	hanks tom	0.436
jackson samuel	0.443	eastwood clint	0.429
lemon jack	0.443	travolta john	0.417
cruise tom	0.435	williams robin	0.417
de niro robert	0.435	douglas michael	0.414
Who are similar to quinn anthony:			
1 level deep		2 levels deep	
banderas antonio	0.366	banderas antonio	21.866
bardem javier	0.350	benigni roberto	20.758
martin steve	0.333	burns george	20.565
goodman john	0.320	baldwin alec	20.413
allen woody	0.285	mcqueen steve	20.244
Who are similar to nicholson jack:			
1 level deep		2 levels deep	
green graham	0.484	banderas antonio	41.477
bronson charles	0.482	bacon kevin	41.133
brody adrien	0.480	baldwin alec	41.0862
bale christian	0.476	bronson charles	40.982
baldwin alec	0.474	benigni roberto	40.948

*friend* es distinto a la similitud de la palabra *friend* a la palabra *family*, esto es por el uso que las personas le dan a las palabras. La palabra *family* y la palabra *friend* no están conectadas directamente, es decir conectadas por una arista, ellas podrían estar conectadas entre vértices intermediarias. Identificar que palabras pueden ser intercambiables es el enfoque del análisis paradigmático.

Nosotros definimos el significante<sup>1</sup> como la similitud entre vértices de un mismo grupo, en que no todos los miembros del grupo tienen el mismo grado de similitud o relación. Esto es por el hecho que un miembro del grupo podría pertenecer a varios grupos con diferentes grados de similitud.

<sup>1</sup>El término significante se utiliza en lingüística estructural y en la semiótica para denominar aquel componente material o casi material del signo lingüístico y que tiene la función de apuntar hacia el significado (representación mental o concepto que corresponde a esa imagen fónica).

#### 4.4.2 Similitud Paradigmatic

PaC usa la equivalencia regular para agrupar vértices. Particularmente el Coeficiente de Similitud Unilateral Jaccard será usado para definir esos vértices que tienen equivalencia regular [STC15b], Unilateral Jaccard (4.7)(4.8).

#### 4.4.3 Algoritmo

En esta sección presentaremos un algoritmo original de agrupamiento PaC, nosotros nos asistiremos en el uso de un grafo bipartito, para poder realizar un rápido y fácil análisis paradigmático. Nuestro enfoque es comparar sólo filas que tienen por lo menos una columna en común, el cual reduce el espacio de comparación. Nuestro abordaje es anclar una fila hacia sus filas más similares. Si queremos extender el nivel de agrupación, podríamos anclar una fila a más de una fila basado en el grado de similitud. PaC algoritmo puede procesar columnas o filas indiferentemente, resultando en bi-clustering algoritmo. PaC da buenos resultados en matrices escasas debido al bajo nivel de conectividad, su no dependencia permite ejecutar PaC de forma paralela.

El grafo de ingreso debe de ser bipartito, si el grafo no es bipartito; nosotros podemos transformar el grafo a un grafo bipartito en el momento del cálculo, el modelo bipartito permitirá tener cada palabra en una fila o en una columna, podemos configurar el conjunto de palabras como columna-palabra-xy y fila-palabra-xy. PaC puede ser aplicado a las filas o columnas indistintamente.

#### 4.4.4 PaC algoritmo

PaC usa el cálculo de modularidad de grafos, el uso de la modularidad no es intensivo, puesto que es un uso de arriba hacia abajo, el uso de la modularidad no es con el fin de identificar los grupos, por el contrario es para optimizar los grupos encontrados. PaC recalculará su umbral, para identificar el corte óptimo de los grupos. En cada iteración de la modularidad puede mejorar los grupos encontrados. Los grupos iniciales podrían ser o no del interés de la aplicación, pero para objetivos generales y comparativos trataremos de optimizar los grupos encontrados. Los grupos dependerán de cuan similar los grupos necesitan ser, modularidad ayudará en encontrar el corte óptimo. El umbral es uno para todos, pero este cambiará según sea el caso en cada grupo, esto quiere decir que cada grupo contará con su propio umbral.

La función de uJaccard ha sido presentado en (4.7)(4.8); y la modularidad de grafos es introducido por Newman-Girvan [NG04] es dado por:

Listing 4.1 PaC algoritmo

```

// The graph needs to be bipartite
// The graph needs to be an undirected graph
procedure PaC (graph, threshold)
for each vertex in the graph
similaritiesPairs[ ] = uJaccard (Vi,Vn,)
// uJaccard two levels out, all possible
// paths among Vi and Vn
sort similaritiesPairs[ ]
group vertices with similar structural
equivalence & uJaccard score.
grouping up to a threshold
clusters [ ] = groups found
return clusters[ ]
// the grouping is done by selecting the elements
// up to a particular threshold

```

Listing 4.2 Algoritmo de Agrupacion

```

// we start with modularity of 0
// we start with a threshold = 0.5
// new modularity is nQ = 0
// last modularity is lQ = -1
// factor will be the step of reduction = 0.01
procedure clus (graph, clusters, threshold)
while nQ > lQ
lQ = nQ
threshold[ ] = threshold - factor
cluster[ ] = PaC(graph, threshold)
nQ = modularity (graph, clusters)
return cluster[ ]
// threshold is a map <cluster id, threshold >
// some clusters will be improved
// based on its assigned threshold

```

$$Q = \sum_{i=1}^k (eii - a_i^2) \quad (4.9)$$

Donde  $eii$  = % aristas en el módulo  $i$  (probabilidad de las aristas es en modulo  $i$ ),  $ai$  = % aristas con al menos 1 en modulo  $i$  (probabilidad aleatoria de que una arista caiga dentro del módulo  $i$ ).

La primera corrida genera un conjunto de grupos, ¿pero estos grupos son los óptimos? Para medir si los grupos generados es el óptimo corte del grafo, la modularidad [NG04] es utilizada dado por (4.9). Modularidad correrá después de PaC, se reducirá el umbral y correrá nuevamente la modularidad, si se logró algún tipo de optimización se iterará caso contrario se detendrá el algoritmo. Se mantendrá un umbral en cada grupo y en cada iteración se revisará si el grupo ha mejorado o no. Si ninguno de los grupos ha mejorado entonces se detendrá el algoritmo caso contrario se iterará nuevamente. Puesto que es un enfoque de arriba hacia abajo el número de iteraciones es mínima, y el uso de la modularidad no es extensivo a diferencia de los modelos de abajo hacia arriba.

La complejidad de ejecución del algoritmo es  $O(n * avg(degree)^2)$ . Podemos apuntar que el algoritmo es altamente paralelizable, puesto que su cálculo de similitud no depende de cálculos previos. Es un algoritmo por flujo, Por último el algoritmo es incremental, puesto que podemos adicionar nuevos vértices y aristas y solo re calcular el área afectada y no todo el grafo.

## 4.5 PaC Evaluación Experimental

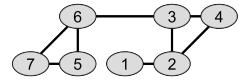
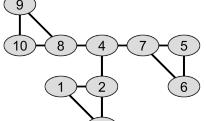
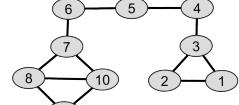
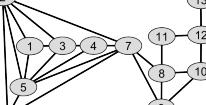
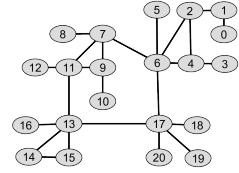
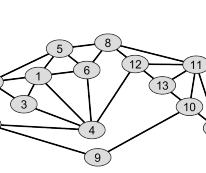
Se ha evaluado el algoritmo en grafos pequeños, conjunto de datos sintéticos, conjunto de datos de Procesamiento del Lenguaje Natural y conjunto de datos reales. Para probar su eficiencia PaC se ha utilizado la información mutua normalizada para compararlo con una serie de algoritmos de agrupación en grafos sintéticos [DDGDA05].

### 4.5.1 Evaluación con grafos pequeños

El uso de PaC en grafos pequeños como se muestra en la Tabla 4.8, es claro observar que los grupos generados por PaC son los deseados, más aún los algoritmos del estado del arte también generan los mismos grupos.

Los pequeños grafos en la Tabla 4.8 sólo son ejemplos simples de la evaluación del algoritmo PaC, particularmente en casos especiales PaC es capaz de identificar los grupos correctamente, como en el tercer grafo en el grupos 4, 5, 6 ha sido identificado correctamente,

Tabla 4.8 Agrupacion de pequeños grafos utilizando PaC

	
(1,2,3,4) (5,6,7)	(1,2,3) (4,8,9,10) (5,6,7)
	
(1,2,3) (4,5,6) (7,8,9,10)	(1,2,3,4,5,6) (7,8,9) (10,11,12,13)
	
(0,1,2,3,4,5,6) (7,8,9,10,11,12) (13,14,15,16) (17,18,19,20)	(1,2,3,4,5,6) (7,8,9,10,11,12,13,14)

así también el ultimo grafo en que dos grupos han sido identificados correctamente como se esperaba.

#### 4.5.2 Grafos reales

**Zachary's karate club:** La red representa los patrones de amistad entre los miembros de un club de Karate de una universidad de Norte América [Zac77].

**Football:** Red football Americano [GN02].

**Dolphins:** Rede social de delfines de Nueva Zelanda [Lus03].

**Les Miserables:** Red de interacción entre los caracteres más importantes de la extensa novela de la delincuencia y la redención de Víctor Hugo, Los Miserables [Knu].

Para medir nuestros resultados con un grafo del que se conoce los grupos, nosotros hemos usado Información Mutua Normalizada (Normalized Mutual Information - NMI) [FH61]. Es una medida de la similitud de partición tomada de la teoría de la información, que ha demostrado ser fiable [DDGDA05]. Utilizada por [For10] para comparar dos grafos y obtener un puntaje de diferencia entre los dos grafos. Si la NMI es igual a 1 esto demuestra que los grafos son iguales, 0 demuestra que los grafos son completamente diferentes.

Nuestros resultados se muestran en la Tabla 4.9, la columna *PaC* es comparada en la columna *C* que muestra los grupos conocidos, la evaluación se muestra en la columna *NMI*, estos resultados demuestran que PaC provee buenos resultados.

Tabla 4.9 Agrupación de redes reales utilizando PaC

	<b>N</b>	<b>E</b>	<b>C</b>	<b>NMI</b>	<b>PaC</b>
<b>Zachary's karate club</b>	34	78	2	0.652	4
<b>Dolphins</b>	62	159	2	0.527	5
<b>Football</b>	115	613	12	0.907	15
<b>Les Miserables</b>	77	254	11	0.878	12

En la tabla 4.9, de izquierda a derecha, el número de vértices N, aristas E, normalizado de información mutua NMI, número real de los grupos C (línea base) y el número grupos generados por PAC.

PaC es capaz de identificar prácticamente de forma correcta los grupos en el grafo de Football con un NMI de 0.9, similarmente en el grafo Les Miserable con un NMI de 0.878, pero no tan bien con el grafo de los delfines donde PaC identificó 5 grupos de los 2 existentes; en el grafo de Zachary PaC identificó 4 grupos de los 5 existentes, pero este es un caso particular porque los grupos conocidos eran 2 pero el estado del arte ha mostrado que existen 4 grupos con algunos vértices que podrían pertenecer a diferentes grupos.

#### 4.5.3 Datos Sintéticos

Se ha evaluado PaC, basado en un análisis comparativo con grafos construidos con grupos conocidos de antemano, estándar Benchmark como el de Girvan and Newman no considera características importantes de grafos reales, como la distribución fat-tailed del grado de vértices y el tamaño de grupos [LFR08]. Por esto, nosotros hemos utilizado una clase diferente de comparación de grafos de red, en el cual la distribución del grado de vértices y el tamaño de los grupos son tomadas en cuenta y configurables en el proceso de construcción de los grafos.

Para comparar los grupos de encontrados por PaC y los grupos conocidos de antemano, nosotros usaremos la Información Mutua Normalizada [For10], para nuestras evaluaciones usaremos la configuración usada en [LF09].

Para validar el desempeño de PaC, PaC se comparó con 6 técnicas presentes en iG-graph [CN06], Walktrap Community (wc), Spinglass Community (sc), Leading Eigenvector Community (lec), Fastgreedy Community (fc), Label Propagation Community (lpc) and Multilevel Community (mc).

El parámetro de mezcla en la figura 4.7 , 4.8 y 4.9 es el grado en el que los grupos se mesclan con otros grupos, 0 significa que los grupos son completamente separados, cuando se usa 0.5 esto significa que el 50% de las aristas en un grupo están conectadas a otros grupos. El parámetro de realización en las figuras 4.7 , 4.8 y 4.9 se refiere al número de evaluaciones

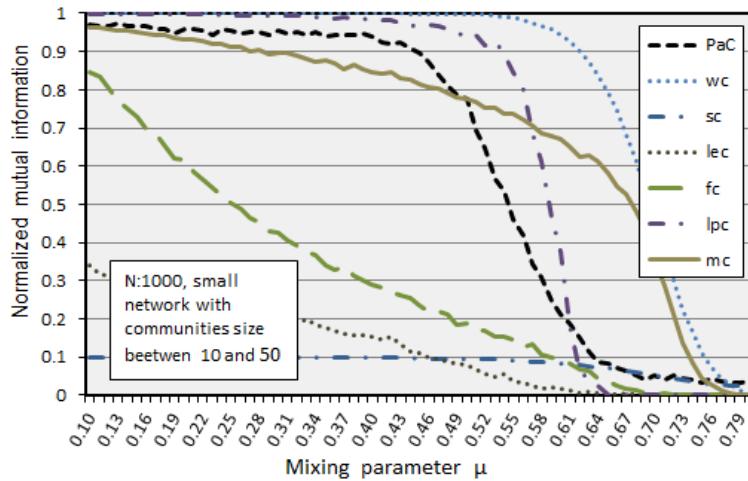


Figura 4.7 Comparación de PaC en datos sintéticos, N=5000

Comparando PaC con los algoritmos, grafo con comunidades entre 10 y 50 vértices; wc, sc, lec, fc, lpc, mc en datos sintéticos, el número de vértices es N=5000, cada punto corresponde a 100 realizaciones de grafos. PaC claramente supera a los algoritmos sc, lec, fc.

se han realizado en cada punto, tales como 1000 , 5000, 1000 evaluaciones por punto. Esto se realiza de esta forma que la generación de los grafos es aleatoria.

PaC tiene un mayor desempeño que Multilevel Community (mc), Fastgreedy Community (fc), Leading Eigenvector Community (lec) y Spinglass Community (sc) pero no es el mismo caso para Walktrap Community (wc) y Label Propagation Community (lpc). En el caso de Walktrap Community (wc) este tiene un tiempo de ejecución cuadrático  $O(n^2 \log n)$  [PL05] el cual no es apropiado para grafos de gran escala, en comparación con PaC que tiene un tiempo de ejecución de  $O(n \times \text{avg(degree)}^2)$ . En el caso de Label Propagation Community (lpc), lpc tiene un tiempo de ejecución similar a PaC, la diferencia en desempeño es por el hecho de que cada algoritmo usa diferente medidas de similitud, mientras que lpc usa una medida simétrica y PaC usa un coeficiente asimétrico, por consiguiente no podemos compararlos ambos sólo basados en el resultado, se debe de considerar también el tipo de medida de similitud utilizada, PaC usa un análisis paradigmático, enfocado al Procesamiento del Lenguaje Natural.

#### 4.5.4 Evaluación NLP

Uno de los desafíos en las áreas de trabajo del Procesamiento del Lenguaje Natural (NLP por sus siglas en inglés) es la desambiguación de sentidos, siendo esta más compleja en un enfoque no supervisado. Nosotros usaremos PaC para atacar este problema y usaremos un análisis empírico para valorar nuestros resultados. Para lograr esto PaC necesitará datos de

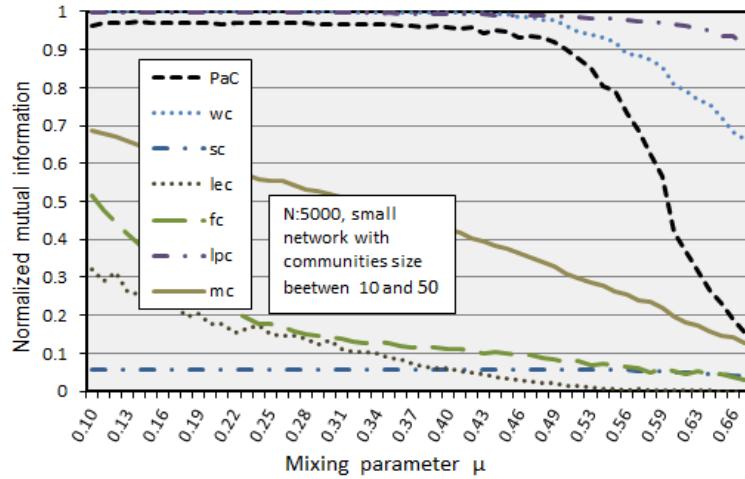


Figura 4.8 Comparación de PaC en datos sintéticos, N=1000

Comparando PaC con los algoritmos, grafos con comunidades entre 10 y 50 vértices; wc, sc, lec, fc, lpc, mc en datos sintéticos, el número de vértices es N=1000, cada punto corresponde a 100 realizaciones de grafos. PaC claramente es superior que sc, lec, fc, mc.

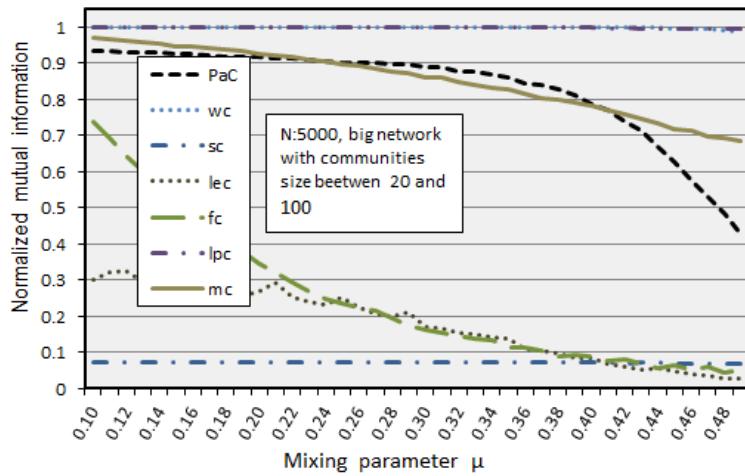


Figura 4.9 Comparación de PaC en datos sintéticos, N=1000, comunidades grandes

Comparando PaC con los algoritmos, grafos con comunidades entre 20 y 100 vértices; wc, sc, lec, fc, lpc, mc en datos sintéticos, el número de vértices es N=1000, cada punto corresponde a 100 realizaciones de grafos. PaC claramente es superior que sc, lec, fc, mc.

Tabla 4.10 Tamaños de los corpus

Corpus	Google bi-gram	BNC	Wikipedia
<b>Size</b>	1.5GB	539MB	11GB
<b>bi-grams</b>	314,843,401	842,162,318	
<b>Words</b>	629,298,853	98,537,224	1,459,026,075
<b>Vocabulary</b>	5,827,772	354,374	3,990,248

ingreso en la forma de un grafo, un grafo donde los vértices son las palabras, con precisión los lemas y las aristas son la relación de uso que tienen estas.

Para construir nuestros datos de entrada, nosotros seleccionamos tres fuentes de datos, Google n-gram corpus [FB06], The British National Corpus, version 3, BNC XML Edition [C<sup>+</sup>07] y en.Wikipedia<sup>2</sup> database backup dumps, cada cuerpo conformará un conjunto separado de evaluación.

Para construir un grafo de palabras se inició construyendo un modelo bi-grama, esto fue muy directo con el dataset de Google n-gram, puesto que este son bi-gramas propiamente dichos y no existió la necesidad de pre-procesarlos, pero si limpiarlos. En el caso de en.Wikipedia y BNC se realizaron una serie de tareas para limpiar los datos, en particular los datos de en.Wikipedia, extraído los datos en formato texto se procedió a identificar las oraciones, a través de un identificador de oraciones, el cuál es basado en una serie de heurísticas. Una vez las oraciones fueron identificadas se analizaron para descubrir las partes de la oración, POS por sus siglas en inglés (part of speech), para esto utilizamos BLLIP Parser con una configuración de T50 [MCJ06]. Una vez analizado las oraciones e identificado las POS de cada oración, se removieron las palabras más comunes (las 1000 palabras más comunes), posterior a esto se obtuvo el lema de cada palabra realizando una vista rápida en WordNet [Fel05a].

La construcción del modelo bi-grama siguió los siguientes pasos, se generaron varios modelos de bi-gramas verb-verb, noun-noun, verb-noun, noun-verb, adverb-verb, verb-adverb, adjective-noun, noun-adjective. Se utilizó una ventana de tamaño  $n$  para desplazarla a la derecha y a la izquierda de la palabra a generar el bi-grama; después de varios experimentos se observó que  $n$  no debería exceder a 3, puesto que el significado de la palabra comienza a perderse. Esto es una extensión al trabajo de [JK95] and [Pan03].

Se seleccionó dos medidas de colocación del estado del arte, Log-likelihood Ratios LLR [Dun93]. LLR ha sido seleccionado por que tiene un buen abordaje a los datos escasos y es capaz de identificar bi-gramas raros con baja frecuencia.

---

<sup>2</sup>versión en inglés

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)} \quad (4.10)$$

$$\log \lambda = \log \frac{b(c_{12}, c_1, p) b(c_2 - c_{12}, N - c_1, p)}{b(c_{12}, c_1, p_1) b(c_2 - c_{12}, N - c_1, p_2)} \quad (4.11)$$

La segunda medida de colocación que seleccionamos es Pointwise Mutual Information (PMI), del área de teoría de la información, mutual información se refiere a información mutua que dos variables aleatorias tienen a diferencia de dos variables relacionadas. Esta es una mala medida de dependencia porque esta no depende de la frecuencia. PMI es mala con datos escasos, algunas palabras que sólo ocurren una vez y aparecen juntas tienen una alta anotación de PMI. PMI también es mala con dependencias de palabras, donde dos palabras son perfectamente dependientes de la otra, cuando una ocurre la otra también ocurre. Esto quiere decir que las palabras más raras obtienen una marcación alta en el PMI. [Bou09] muestra varios modelos de normalización para PMI, para lograr que sea menos sensitiva a las frecuencias bajas, de los varios modelos presentados en [Bou09] y realizado algunas evaluaciones, se seleccionó la normalizada de PMI a la potencia de cubo (Normalized Pointwise Mutual Information to the power of cube)  $NPMI^3$ , puesto que reduce la marcación de palabras raras de forma apropiada en grandes cantidades de datos.

$$PMI(x, y) = \log \frac{p(x, y)}{p(x)p(y)} \quad (4.12)$$

Se calculó  $LLR$  y  $NPMI^3$  por cada bi-grama, dado una palabra se alineó los resultados de  $LLR$  y  $NPMI^3$ , formando un vector característico de la palabra dada. Sólo trabajamos con  $1/3$  del vector característico puesto que su marcación de colocación comienza a ser muy baja e insignificante a partir de  $2/3$ . Es interesante notar que en muchos casos el vector característico de una palabra tiene valores distintos para  $LLR$  y  $NPMI^3$  pero el orden de la palabras es el mismo o las variaciones son mínimas. Esto da una clara indicación que  $LLR$  y  $NPMI^3$  están trabajando como se esperaba.

Se construyó una serie de grafos de palabras, uno por cada caso (verb-verb, noun-noun, verb-noun, noun-verb, adverb-verb, verb-adverb, adjective-noun, noun-adjective) por cada repositorio BNC, en.Wikipedia y Google n-gram.

Para la desambiguación de palabras no es necesario correr PaC en todo el grafo, por el contrario se obtuvo un subconjunto de palabras relacionadas de la palabra dada. Este sub-grafo se construye de la siguiente forma: se obtiene el vector característico de la palabra dada y se carga en el grafo, luego por cada palabra en el vector característico se carga el vector característico de la palabra relacionada y se carga en el grafo, de forma similar a

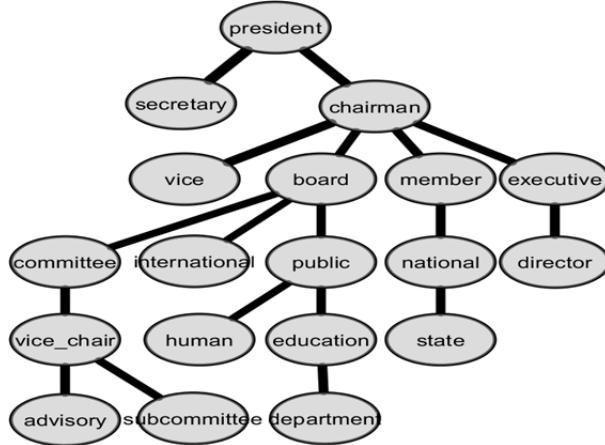


Figura 4.10 Plabra "chair" de en.wikipedia

La palabra "chair" se extrajo del repositorio en.wikipedia (noun-noun), se muestran los grupos mas grandes.

[FB04]. Luego agrupamos el sub-grafo utilizando PaC, obteniendo grupos que representan sentidos de la palabra dada.

PaC produce grupos que representan sentidos de las palabras, como se muestra en la Figura 4.9 a la 4.12 y Tablas 4.11 y 4.12, el Coeficiente de Similitud Unilateral Jaccard ha demostrado cuan bien relacionado están las palabras en cada grupo. El grosor de las aristas demuestra el grado de similitud entre ambas palabras, el cual es generado por uJaccard. Podemos Observar que cada corpus genera diferentes grupos de una misma palabra, esto se debe a que los contextos son distintos, como en el caso de *chair*. Los resultados son meritorios de considerar. Si consideramos comparar los resultados con un corpus reconocido como el de WordNet u otro repositorio, este sería muy confuso y nada productivo porque cada corpus es construido de forma distinta y tiene distintos contextos según argumenta Aguirre [AADL04].

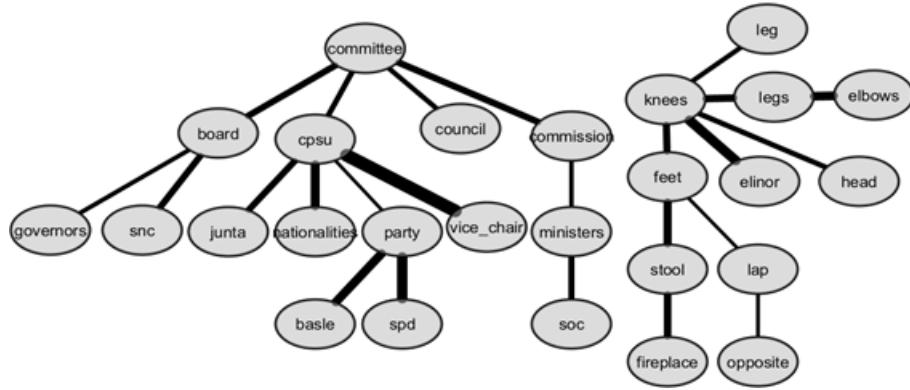


Figura 4.11 Palabra "chair" de BNC.

La palabra "chair" se extrajo del repositorio BNC (noun-noun), se muestran los grupos más grandes.

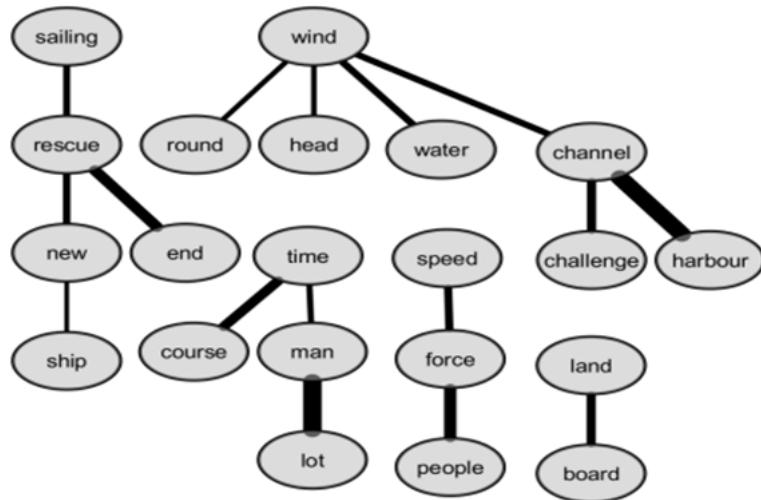


Figura 4.12 Palabra "sail" de BNC

La palabra "sail" se extrajo del repositorio BNC (noun-verb), se muestra los grupos mas grandes.

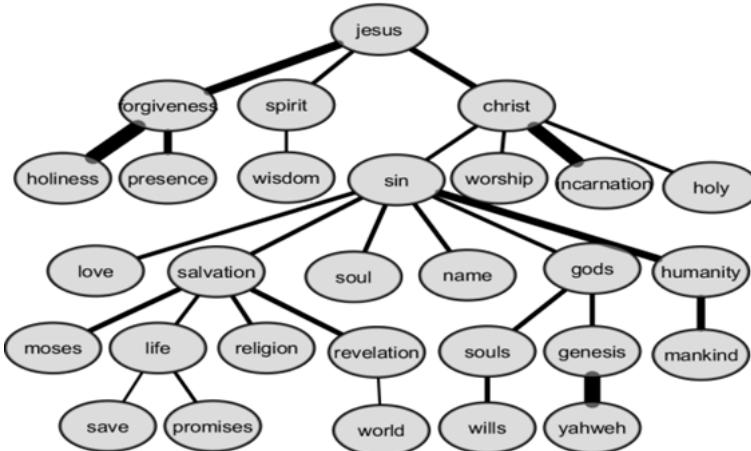


Figura 4.13 Palabra "god" de BNC

La palabra "god" se extrajo del repositorio BNC (noun-noun), se muestran los grupos mas grandes.

Tabla 4.11 Grupos o sentidos de la palab "apple".

Grupos que representan los sentidos de la palabra "apple", producido por PaC utilizando el repositorio en.wikipedia. El grupo relacionado a la fruta es presente y otros grupos agrupan vértices basado en sus sentidos de la pala "apple" tales como música.

macintosh , os, macos, mac, desktop
notebook , laptop, asus, baterry
wireless, usb, memory, remote
release , announce, calendar, make
store, product, user
imac , emac, intel, macbook
cranberry , pectin, cinnamon
juice, cider, martini
software, final, pro, power, hardware, advisor, video, quicktime
music, cinema, aria, garage band, book, deal, vacation, dvd, lossless, itunes, ipod, insight, ipods, archos
valley, spring, orchard, avery, tree, blossom, newton, loop, airport
support, core, developer, weblog, inspiration, iie, iigs
powerbook, ibook, powermac, xserve

Tabla 4.12 Grupos o sentidos de la palabra "god".

Grupos que representan los sentidos de la palabra "god", producido por PaC utilizando el repositorio de Google bi-grams. Es claro observar que cada grupo representa un sentido los distintos sentidos que la palabra "god" tiene.

punish, forgive, almighty, forbid, instruct, spake, ordain
play, free, man, smile, soldier, live, tv
incarnate, bless, forsaken, sake, fear, manifest
told, knew, father, hear
send, save, word, rulesnet
damned, damn, awful, ente
mode, speed, command, channel
brought, bring, giveth, meant, reveal, bring, make, awfuls, gave
desire, set, choose, expect, require, dwell, change, raise, intend, accord
government, church, grant, design, exist, heal, lead

PaC produce grupos que representan los sentidos de una palabra, como se muestra en la Tabla 4.11 y 4.12, el Coeficiente de Similitud Unilateral Jaccard demuestra cuán bien están relacionadas las palabras en el corpus Google bi-grams. Las palabras en el corpus Google bi-grams no tienen ningún tipo de información sintáctica o POS, sólo se utilizó las frecuencias de las palabras, *LLR*, *NPMI* y PaC, esta es una observación muy interesante que a pesar de la falta de información sintáctica y POS los grupos generados por PaC son muy significativos.

## 4.6 Conclusion

En esta tesis se presenta un novedoso algoritmo de agrupación PaC, el cual identifica grupos o comunidades basado en la propiedad paradigmática y el Coeficiente de Similitud Unilateral Jaccard. PaC es altamente paralelizable, incremental y es un algoritmo de agrupación en flujo, el cual es apropiado para gran cantidad de datos. PaC tiene una complejidad de ejecución de  $O(n \times \text{avg}(\text{degree})^2)$ . Los resultados muestran que PaC es capaz de encontrar los sentidos de una palabra, así como agrupar sin supervisión alguna.

En esta tesis se ha presentado una serie de evaluaciones de diferentes casos, los cuales han confirmado la usabilidad de uJaccard. Nosotros podríamos extender uJaccard a través de inclusión de pesos o valores ponderados, con el objetivo de mejorar la asimetría. Trabajo que se presenta en el siguiente capítulo.



# Capítulo 5

## Coeficiente Ponderado de Similitud Unilateral Jaccard

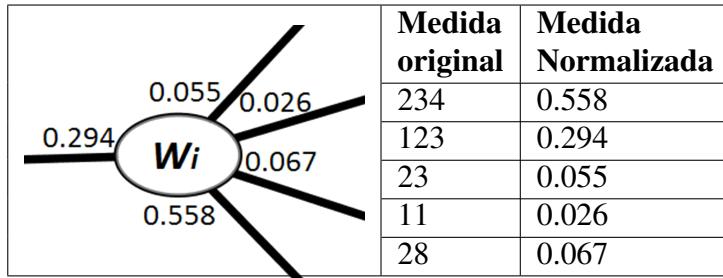
---

### 5.1 Introducción

Medidas de similitud son esenciales para resolver muchos problemas de reconocimiento de patrones como los de clasificación, agrupamiento, extracción de información entre otros. Las medidas de similitud son categorizadas en modelos de relación sintáctica y modelos de relación semántica. En esta tesis se presenta un coeficiente de similitud novedoso y original, Coeficiente Ponderado de Similitud Unilateral Jaccard, uwJaccard, el cuál toma en consideración no sólo el espacio entre dos puntos sino también la relación semántica entre los puntos en un modelo de distribución semántica, el Coeficiente Ponderado de Similitud Unilateral Jaccard, una medida de incertidumbre el cual mide la incertidumbre entre dos oraciones, como *man bites dog* y “*dog bites man*”.

### 5.2 Coeficiente Ponderado de Similitud Unilateral Jaccard

Los pesos para un modelo semántico de distribución, DSM por sus siglas en inglés (distributed semantic model) es un problema abierto como lo muestra Grefenstette y Sadrzadeh [GS15], siendo parte fundamental de un DSM. Es claro que la construcción de un modelo de pesos está relacionado a su contexto, así que no se pueden comparar dos DSMs. En esta tesis presentamos un proceso para generar pesos para un DSM y presentamos el Coeficiente Ponderado de Similitud Unilateral Jaccard.

Tabla 5.1 Vector de características Normalizadas de una palabra  $W_i$ 

Considérese un grafo de palabras, donde los vértices son las palabras y las aristas la relación que tienen entre ellas. Dos palabras son directamente conectadas,  $W_1$  conectada a  $W_2$ . Cada conexión está formada por un peso, se considerará el peso como si fuesen las frecuencias o probabilidades o medida de colocación, de tal forma la probabilidad  $P(W_1, W_2), P(W_2, W_1), P(W_1, W_n)$  y  $P(W_2, W_n)$ . Consecuentemente aristas entre vértice  $W_1$  y vértice  $W_2$  tendrán dos pesos. Puesto que vértice  $W_1$  está conectado a  $W_2$  con una probabilidad o medida de colocación particular, direccional; así también  $W_2$  está conectado a  $W_1$  con una probabilidad o medida de colocación diferente. Por lo tanto, terminamos con una arista entre vértice/palabra  $W_1$  y vértice/palabra  $W_2$  con dos medidas de colocación o medidas de probabilidad. Cabe mencionar que la medida de colocación o probabilidad entre ambos objetos no es simétrica es asimétrica, no es lo mismo *red – car* que *car – red*. De este punto en adelante trabajaremos con medidas de colocación ya que brindan mejor información que una probabilidad. Así también llamaremos a este peso de la arista.

Una palabra es representada por un vector en un grafo, la palabra tendrá un vector característico que representan las palabras a las que es más similar. El vector característico tendrá una medida que está relacionada solo a la palabra dada. El vector característico es muy extenso y es necesario truncarlo para poder relacionar las palabras con mayor significancia y obtener las palabras más características de la palabra dada, el cual puede ser alcanzado a través de diferentes mecanismos, umbral, porcentaje, mínimos o máximos.

Una vez se ha obtenido un vector característico de tamaño maderable, se debe de normalizar las medidas, de tal forma que la *suma* de las medidas es igual a 1, como se muestra en la Tabla 5.1. Por lo tanto, la arista entre dos palabras  $W_1$  y  $W_2$  tendrán dos pesos normalizados, como se muestra en la Figura 5.1. Sin embargo, cada peso normalizado está relacionado a su propia palabra. Por tanto podemos concluir que  $Peso(W_1, W_2)$  and  $Peso(W_2, W_1)$  pertenecen a diferentes modelos y no están relacionadas. Para resolver este problema se propone el uso del Coeficiente Ponderado de Similitud Unilateral Jaccard.

Para calcular Coeficiente Ponderado de Similitud Unilateral Jaccard se consideran dos supuestos. Primero la entropía de la información del vértice es calculada basado en los pesos

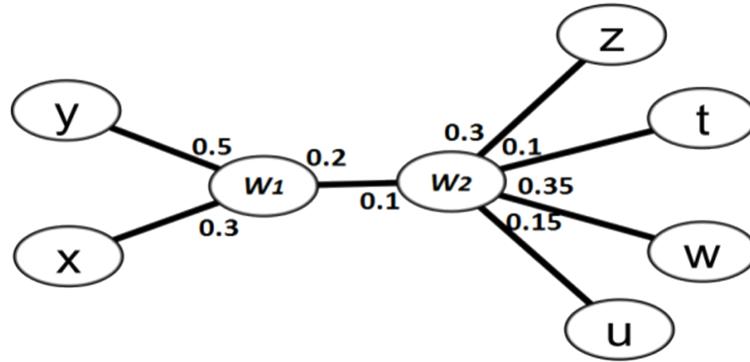


Figura 5.1 Aristas entre palabras  $W_1$  y  $W_2$  con pesos.

Aristas entre la palabra  $W_1$  y la palabra  $W_2$  con pesos de 0.2 y 0.1 respectivamente.

del vector característico de la palabra dada (5.1). Se considerará el vector de características como la distribución de probabilidad de la palabra dada. Para una variable aleatoria discreta  $X$  que puede tomar valores  $x_1, \dots, x_n$  con una probabilidad  $p_i$  respectivamente, define la entropía  $H(X)$  de  $X$ . Donde  $X$  es la palabra y la probabilidad  $p_i, \dots, p_n$  es el vector característico con pesos. La entropía de la información está dada por:

$$H(X) = - \sum_{i=1}^n p_i * \log p_i \quad (5.1)$$

Segundo, nosotros usaremos la entropía cruzada, “cross-entropy” (5.2) para definir la similitud entre  $\text{sim}(W_1, W_2)$  y  $\text{sim}(W_2, W_1)$  como lo presenta Jurafsky, Martin (2009). La entropía cruzada es también conocida como la entropía relativa de  $p$  con respecto de  $q$  y la entropía relativa de  $q$  con respecto a  $p$ , por tanto la entropía cruzada no es simétrica. Donde  $x$  un elemento común en la distribución de  $q$  y  $p$ ,  $P(x)$  es la probabilidad del elemento  $x$  en la distribución  $p$ ,  $Q(x)$  es la probabilidad del elemento  $x$  en la distribución  $q$ .  $x$  es un elemento común en la distribución  $p$  y  $q$ . Por tanto la entropía cruzada es dada solo en las arista enlazando las palabras  $W_1$  y palabra  $W_2$ . Por consiguiente, el Coeficiente Ponderado de Similitud Unilateral Jaccard sólo considera  $x$  como un enlace entre palabras  $W_1$  y palabra  $W_2$ . La entropía cruzada es dada por:

$$H(P, Q) = - \sum_{i=1}^n P(i) \log Q(i) \quad (5.2)$$

El Coeficiente Ponderado de Similitud Unilateral Jaccard,  $uwJaccard$  (5.3),(5.4) es la suma de las entropías de los vértices  $H(W_1)$  y la entropía cruzada en la arista  $H(W_1, W_2)$ . Tabla 5.2 muestra el Coeficiente Ponderado de Similitud Unilateral Jaccard,  $uwJaccard(W_1, W_2)$

Tabla 5.2 uwJaccard ( $W_1, W_2$ )

Coeficiente Ponderado de Similitud Unilateral Jaccard entre las palabras  $W_1$  y word  $W_2$ .

$H(W_1) = 1.4855$
$H(W_2) = 2.1261$
$H(W_1, W_2) = 0.6644$
$H(W_2, W_1) = 0.2322$
$uwJaccard(W_1, W_2) = 2.1499$
$uwJaccard(W_2, W_1) = 2.3583$

y  $uwJaccard(W_2, W_1)$  para la arista entre palabras  $W_1$  y  $W_2$  de la Figura 5.1. El Coeficiente Ponderado de Similitud Unilateral Jaccard está dado por:

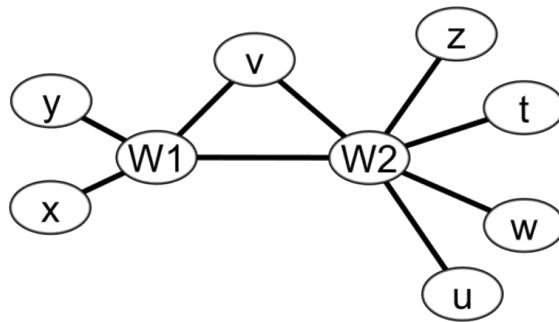
$$uwJaccard(p, q) = H(p) + H(p, q) \quad (5.3)$$

$$uwJaccard(q, p) = H(q) + H(q, p) \quad (5.4)$$

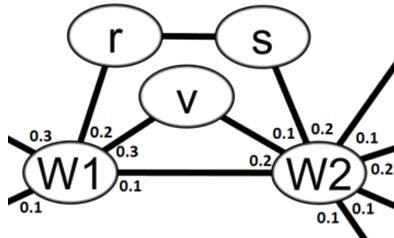
Puesto que no podemos medir distintos modelos de distribución, nosotros argumentamos que es mejor observar cuan incierta es la palabra  $W_1$  de la palabra  $W_1$  y cuan incierta es la palabra  $W_2$  de la palabra  $W_1$ . El Coeficiente Ponderado de Similitud Unilateral Jaccard mide la incertidumbre de la similitud entre palabras  $W_1, W_2$ ;  $uwJaccard(W_1, W_2)$  y  $uwJaccard(W_2, W_1)$ , donde ambos incertidumbres son distintas. El Coeficiente Ponderado de Similitud Unilateral Jaccard es un valor positivo, donde 0 representa una similitud perfecta y cualquier valor mayor a 0 significa incertidumbre.

En consecuencia, el Coeficiente Ponderado de Similitud Unilateral Jaccard no es un peso por el contrario mide el contenido de la información, de ninguna manera no es una medida de similitud más bien es un coeficiente de incertidumbre entre dos palabras.

La similitud no es considerada solo para enlaces directos, pero también para enlaces indirectos, donde existen vértices intermediarios, en el caso de la existencia de multiples caminos desde la palabra  $W_1$  a la palabra  $W_2$ , como se muestra en la Figura 5.2b. El Coeficiente Ponderado de Similitud Unilateral Jaccard (5.5)(5.6) es ajustado para considerar los multiples caminos entre la palabra  $W_1$  y la palabra  $W_2$ . Se debe de considerar todos los posibles caminos y sus segmentos, el cual llegaría a ser muy complejo y complicado de llevar a cabo, es por esto que presentamos un enfoque simplificado. Se suma los caminos al cálculo, por cada camino de la palabra  $W_1$  a la palabra  $W_2$  se calcula la entropía cruzada pero solo se considera la probabilidad saliente de la palabra  $W_1$  y la palabra  $W_2$ , la entropía



(a) Caminos sin pesos



(b) Camino con pesos

Figura 5.2 Aristan entre palabras  $W_1$  y  $W_2$  con múltiples caminos

Aristas entre palabras  $W_1$  y palabra  $W_2$  con múltiples caminos.

cruzada es multiplicada por  $1 + \lambda$ , donde  $\lambda$  representa el número de vértices en el camino, el cual aumentara o disminuirá la incertidumbre de acuerdo con el número de caminos. El Coeficiente Ponderado de Similitud Unilateral Jaccard está dado por:

$$uwJaccard(p, q) = H(p) - H(pO) \sum_{\text{path } \exists p \leftrightarrow q} (1 + \lambda)H(p, q) \quad (5.5)$$

$$uwJaccard(q, p) = H(q) - H(qO) \sum_{\text{path } \exists q \leftrightarrow p} (1 + \lambda)H(q, p) \quad (5.6)$$

Donde  $\lambda = 0.1$  multiplicado por el número de vértices intermedias en el camino,  $H(p, q)$  y  $H(q, p)$  es la entropía cruzada de un camino dado,  $H(p)$  y  $H(q)$  es la entropía del vértice origen.  $H(pO)$  y  $H(qO)$  es la entropía de los enlaces de salida del vértice de salida. El camino  $\exists q \leftrightarrow p$  son todos los caminos posibles que existe entre  $q$  y  $p$ , el camino puede ser 1 o  $n$ , pero para tareas relacionadas a NLP, procesamiento de lenguaje natural, no debe ser mayor a 3, porque se comienza a perder significado.  $\lambda$  podría incrementar la incertidumbre, mayor la cantidad de vértices en camino, mayor es el decremento de similitud, mientras que existen varios caminos, la similitud aumenta, por lo tanto, reducir la incertidumbre por que las palabras están más acopladas.

Tabla 5.3 uwJaccard (  $W_1$ ,  $W_2$  ) con diferentes caminos.

Coeficiente Ponderado de Similitud Unilateral Jaccard entre las palabras  $W_1$  y word  $W_2$ , con diferentes caminos según el grafo en la figura 4b.

$H(W_1) = 2.170950594$
$H(W_2) = 2.721928095$
$H(W_1 \text{ outbound}) = 1.317668107$
$H(W_2 \text{ outbound}) = 1.260964047$
$H(W_1\text{-}R\text{-}S\text{-}W_2) = 0.557262743$
$H(W_1\text{-}V\text{-}W_2) = 1.096236271$
$H(W_1\text{-}W_2) = 0.232192809$
$H(W_2\text{-}R\text{-}S\text{-}W_1) = 0.557262743$
$H(W_2\text{-}V\text{-}W_1) = 0.191066215$
$H(W_2\text{-}W_1) = 0.664385619$
<b>Including 3 paths</b>
$uwJaccard(W_1, W_2) = 3.661341986$
$uwJaccard(W_2, W_1) = 3.756353458$
<b>Including 2 paths</b>
$uwJaccard(W_1, W_2) = 3.367612865$
$uwJaccard(W_2, W_1) = 3.451283524$
<b>Including a direct paths</b>
$uwJaccard(W_1, W_2) = 2.403143404$
$uwJaccard(W_2, W_1) = 3.386313714$

El número de caminos podrían introducir el concepto de acoplamiento, puesto que el número conexiones aumenta, pero esto también introduce incertidumbre. Existe menos incertidumbre cuando el enlace es directo, cuando el número de vértices intermedias en el camino entonces el aumenta la incertidumbre. Esto significa que la similitud disminuye aún más cuando introducimos más rutas con vértices en el medio.

### 5.2.1 Construcción del Grafo de Palabras

Se requiere de un grafo de palabras, para esto se siguió los pasos presentados en el Capítulo 4, sección 4.5.4.

### 5.2.2 Incertidumbre de “man bites dog” y “dog bites man”

En los corpus de Google Bi-gram y en.wikipedia, se muestra la relación entre *dog man* tiene menor incertidumbre 1.1715 y 1.1126 respectivamente, pero en el corpus BNC se muestra lo contrario que la relación *man dog* es menos incierta 1.4038 que *dog man* 2.8767. El corpus de Google bi-grm es de enfoque genérico y se espera que tenga errores, mientras que el corpus de en.wikipedia cubre un numero largo de tópicos y podríamos argumentar que el contexto es muy largo, por otro lado el corpus de BNC se enfoca en interacciones humanas. Por lo tanto podemos concluir que el corpus de Google, el corpus en.wikipedia claramente predicen la diferencia entre *man – dog* y *dog – man* asistidos con el Coeficiente Ponderado de Similitud Unilateral Jaccard.

Para resolver la similitud entre “*man bites dog*” y “*dog bites man*”, primero se calcula la incertidumbre utilizando uwJaccard, de izquierda a derecha siguiendo el trabajo realizado por Grefenstette and Sadrzadeh [GS15]. Podemos encontrar que “*man bites dog*” es más incierto que “*dog bites man*” como se muestra en la Tabla 5.4.

## 5.3 Conclusion

Es difícil comparar similitudes asimétricas con medidas de similitud simétricas, no existe un fuero común de comparación, una alternativa es compara uJaccard y uwJaccard con la similitud de Tversky [Tve77a], pero para esto, se necesita un estímulo, en el caso de Tversky se requiere un juicio humano. Más aún Tversky no considera el concepto de manimos y relaciones indirectas entre dos objetos. Por otro lado, uwJaccard si considera el concepto de caminos y relaciones indirectas entre dos objetos en un grafo.

En esta tesis se presenta un novedoso Coeficiente Ponderado de Similitud Unilateral Jaccard, uwJaccard, el cual es un coeficiente de similitud asimétrica, el cual mide la simili-

Tabla 5.4 Incertidumbre entre *man* y *dog*

Incertidumbre entre palabras *man* and *dog*, utilizando 3 diferentes corpus.

<b>en.wikipedia</b>
H(dog) = 1.112468126, H(man) = 2.902715177 H(dog,man) = 0.000188320 H(man,dog) = 0.544818488 uwJaccard (dog,man) = 1.112656446 uwJaccard (man,dog) = 3.447533665
<b>BNC</b>
H(dog) = 2.870717650 H(man) = 1.393004613 H(dog,man) = 0.006037466 H(man,dog) = 0.010817212 uwJaccard (dog,man) = 2.876755116 uwJaccard (man,dog) = 1.403821825
<b>googleBigram</b>
H(dog) = 1.178393207 H(man) = 1.972395788 H(dog,man) = 0.000640365 H(man,dog) = 0.000857728 uwJaccard (dog,man) = 1.171582092 uwJaccard (man,dog) = 1.933390307

Tabla 5.5 “*man bites dog*” o “*dog bites man*”

incertidumbre entre “*man bites dog*” y “*dog bites man*” basado en el repositorio de datos de Google bigram.

<b>googleBigram</b>
H(man) = 1.972395788
H(dog) = 1.178393207
H(bite) = 2.608230542
uwJaccard(man,bite) = 1.972449690
uwJaccard(bite,dog) = 3.011640027
uwJaccard(dog,bite) = 1.179965631
uwJaccard(bite,man) = 2.692521391
man-bites-dog = 4.984089717
dog-bite-man = 3.872487022

itud basada en la entropía de dos palabras en un grafo. La similitud es la inversa de la incertidumbre entre dos palabras  $W_1$  y  $W_2$ . Se basa en la entropía cruzada para encontrar la incertidumbre, siendo la incertidumbre el coeficiente.  $uwJaccard(word_1, word_2) \neq uwJaccard(word_2, word_1)$ .

También demostramos que existe menor incertidumbre en la oración “*dog bites man*” que la oración “*man bites dog*” la cual es calculada por uwJaccard.



# **Capítulo 6**

## **Conclusiones**

---

### **6.1 Introducción**

El presente trabajo de investigación doctoral tiene 4 aportes al estado del arte:

1. Nuevo Coeficiente de Similitud Unilateral Jaccard, el cual es una extensión del coeficiente de Jaccard e incorpora el concepto de similitud asimétrica entre dos objetos.
2. Nueva propiedad Paradigmática para grafos, el cual es definido como el conjunto de vértices que son similares si ellos tienen una equivalencia regular y son intercambiables por sustitución o transposición.
3. Nuevo algoritmo de agrupación PaC (Paradigmatic Clustering) para grafos.
4. Nuevo Coeficiente Ponderado de Similitud Unilateral Jaccard, el cual es un coeficiente de incertidumbre e incorpora el concepto de incertidumbre asimétrica entre dos objetos.

### **6.2 Conclusiones**

Un elemento clave que se asume en la gran mayoría de los modelos de similitud es que la relación simétrica de similitud. El supuesto de simetría no es universal, y no es esencial para todas las aplicaciones de similitud. La necesidad de la similitud asimétrica es importante y central en la extracción de información, redes de datos y grafos. Este puede mejorar métodos actuales y proveer otro punto de vista.

En esta tesis se está presentando un novedoso coeficiente simétrico de similitud, el Coeficiente de Similitud Unilateral Jaccard,  $uJaccard$ , donde la similitud entre  $a$  y  $b$  no es la misma que la similitud entre  $b$  y  $a$ ,  $uJaccard(A, B) \neq uJaccard(B, A)$ , el cual está basado en el análisis paradigmático, en comparación a Tversky [Tve77b] nuestro enfoque no requiere un estímulo generado por un ser humano, el cual es el caso de Tversky.

Se ha podido demostrar como la detección de grupos o comunidades puede ser enfocado desde un punto de vista paradigmático y utilizando un coeficiente de similitud asimétrico. En el cual, la idea principal es identificar cuan similar son dos objetos no adyacentes. El análisis paradigmático es el proceso que identifica entidades que no son relacionadas directamente, pero son similares porque sus propiedades y sus relaciones a otras entidades lo son, donde son regularmente equivalentes.

Así también en esta tesis se introduce una nueva propiedad de grafos, la propiedad paradigmática, la cual es definida como el conjunto de vértices que son similares si ellos tienen una equivalencia regular y son intercambiables por sustitución o transposición.

En esta tesis se presenta un novedoso algoritmo de agrupación PaC, el cual identifica grupos o comunidades basado en la propiedad paradigmática y el Coeficiente de Similitud Unilateral Jaccard. PaC es altamente paralelizable, incremental y es un algoritmo de agrupación en flujo, el cual es apropiado para gran cantidad de datos. PaC tiene una complejidad de ejecución de  $O(n * avg(degree)^2)$ . Los resultados muestran que PaC es capaz de encontrar los sentidos de una palabra, así como agrupar sin supervisión alguna.

En esta tesis se presenta un novedoso Coeficiente Ponderado de Similitud Unilateral Jaccard,  $uwJaccard$ , el cual es un coeficiente de similitud asimétrica, el cual mide la similitud basada en la entropía de dos palabras en un grafo. La similitud es la inversa de la incertidumbre entre dos palabras  $W_1$  y  $W_2$ . Se basa en la entropía cruzada para encontrar la incertidumbre, siendo la incertidumbre el coeficiente.  $uwJaccard(word_1, word_2) \neq uwJaccard(word_2, word_1)$ .

### 6.3 Trabajos Futuros

Uno de los aportes más importantes de la tesis es poder introducir medidas asimétricas al proceso de agrupamiento en grafos y en forma general al área de inteligencia artificial. Se está trabajando en el uso de  $uJaccard$  en la segmentación de imágenes, el cual podrá segmentar figuras irregulares y entrelazadas. Se espera incorporar  $uJaccard$  en diferentes áreas.

Otro trabajo en el cual estamos trabajando es el uso de PaC en un contexto de agrupación paralela en modelo de sistemas distribuido con flujo de datos continuo e incremental.

# Bibliografía

- [AADL04] Eneko Agirre, Enrique Alfonseca, and Oier Lopez De Lacalle. Approximating hierarchy-based similarity for wordnet nominal synsets using topic signatures. In *Proceedings of GWC-04, 2nd global WordNet conference*, pages 15–22, 2004.
- [ABKS99] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. In *ACM Sigmod Record*, volume 28, pages 49–60. ACM, 1999.
- [Abn07] Steven Abney. *Semisupervised learning for computational linguistics*. CRC Press, 2007.
- [Att55] Fred Attneave. Symmetry, information, and memory for patterns. *The American journal of psychology*, 68(2):209–222, 1955.
- [BB98] Amit Bagga and Breck Baldwin. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer, 1998.
- [BC57] J Roger Bray and John T Curtis. An ordination of the upland forest communities of southern wisconsin. *Ecological monographs*, 27(4):325–349, 1957.
- [Bez13] James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.
- [BKA02] Henrik Bulskov, Rasmus Knappe, and Troels Andreasen. On measuring similarity for conceptual querying. In *International Conference on Flexible Query Answering Systems*, pages 100–111. Springer, 2002.
- [BMM07] Sanghamitra Bandyopadhyay, Anirban Mukhopadhyay, and Ujjwal Maulik. An improved algorithm for clustering gene expression data. *Bioinformatics*, 23(21):2859–2865, 2007.
- [Bou09] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009.
- [BP07] Sanghamitra Bandyopadhyay and Sankar Kumar Pal. *Classification and learning using genetic algorithms: applications in bioinformatics and web intelligence*. Springer Science & Business Media, 2007.

- [Bri98] Derek Bridge. Defining and combining symmetric and asymmetric similarity measures. In B. Smyth and P. Cunningham, editors, *Advances in Case-Based Reasoning (Procs. of the 4th European Workshop on Case-Based Reasoning)*, LNAI 1488, pages 52–63. Springer, 1998.
- [BS07] Sanghamitra Bandyopadhyay and Sriparna Saha. Gaps: A clustering method using a new point symmetry-based distance measure. *Pattern recognition*, 40(12):3430–3451, 2007.
- [BS08] Sanghamitra Bandyopadhyay and Sriparna Saha. A point symmetry-based clustering technique for automatic evolution of clusters. *IEEE Transactions on Knowledge and Data Engineering*, 20(11):1441–1457, 2008.
- [BS12] Sanghamitra Bandyopadhyay and Sriparna Saha. *Unsupervised classification: similarity measures, classical and metaheuristic approaches, and applications*. Springer Science & Business Media, 2012.
- [C<sup>+</sup>07] BNC Consortium et al. The british national corpus, version 3 (bnc xml edition). distributed by oxford university computing services pp the bnc consortium, 2007.
- [CCGV10] Javier Tejada Cárcamo, Hiram Calvo, Alexander Gelbukh, and José Villegas. Impacto de recursos léxicos manuales y automáticos en la wsd. CLEI, 2010.
- [CJ10] Zheng Chen and Heng Ji. Graph-based clustering for computational linguistics: A survey. In *Proceedings of the 2010 workshop on Graph-based Methods for Natural Language Processing*, pages 1–9. Association for Computational Linguistics, 2010.
- [CN06] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5):1–9, 2006.
- [CSL02] Chien-Hsing Chou, Mu-Chun Su, and Eugene Lai. Symmetry as a new measure for cluster validity. In *2nd WSEAS Int. Conf. on Scientific Computation and Soft Computing*, pages 209–213, 2002.
- [DD09] Michel Marie Deza and Elena Deza. Encyclopedia of distances. In *Encyclopedia of Distances*, pages 1–583. Springer, 2009.
- [DDGDA05] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.
- [DHC09] Hai Dong, Farookh Khadeer Hussain, and Elizabeth Chang. A hybrid concept similarity measure model for ontology environment. In *OTM Confederated International Conferences "On the Move to Meaningful Internet Systems"*, pages 848–857. Springer, 2009.
- [DHS] RO Duda, PE Hart, and DG Stork. Pattern classification 2nd ed., 2001.
- [DK82] Pierre A Devijver and Josef Kittler. *Pattern recognition: A statistical approach*. Prentice hall, 1982.

- [DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [dS59] F de Saussure. Course in general linguistics. 1959.
- [DSB11] Ferdinand De Saussure and Wade Baskin. *Course in general linguistics*. Columbia University Press, 2011.
- [Dun93] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74, 1993.
- [EKS<sup>+</sup>96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.
- [ELL01] Brian S Everitt, S Landau, and M Leese. Cluster analysis arnold. *A member of the Hodder Headline Group, London*, 2001.
- [Eve78] Brian S Everitt. *Graphical techniques for multivariate data*. North-Holland, 1978.
- [FB04] Xiaoli Zhang Fern and Carla E Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*, page 36. ACM, 2004.
- [FB06] Alex Franz and Thorsten Brants. All our n-gram are belong to you. *Google Machine Translation Team*, 20, 2006.
- [FC12] Santo Fortunato and Claudio Castellano. Community structure in graphs. In *Computational Complexity*, pages 490–512. Springer, 2012.
- [Fel98] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.
- [Fel05a] C Fellbaum. Wordnet and wordnets in k. brown (ed.), encyclopedia of language and linguistics (pp. 665–670), 2005.
- [Fel05b] Christiane Fellbaum. Wordnet and wordnets. In Alex Barber, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier, 2005.
- [FH61] Robert M Fano and David Hawkins. Transmission of information: A statistical theory of communications. *American Journal of Physics*, 29(11):793–794, 1961.
- [For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.
- [GB02] Peter Grabusts and Arkady Borisov. Using grid-clustering methods in data classification. In *PARELEC*, volume 2, page 425, 2002.
- [GMW07] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*, volume 20. Siam, 2007.

- [GN02] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.
- [GS15] Edward Grefenstette and Mehrnoosh Sadrzadeh. Concrete models and empirical evaluations for the categorical compositional distributional model of meaning. *Computational Linguistics*, 41(1):71–118, Feb 2015.
- [Ham50] Richard W Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.
- [HD79] Frederick Hartwig and Brian E Dearing. *Exploratory data analysis*. Sage, 1979.
- [HdCC04] Eduardo R Hruschka, Leandro Nunes de Castro, and Ricardo JGB Campello. Evolutionary algorithms for clustering gene-expression data. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 403–406. IEEE, 2004.
- [Hol79] Eric W Holman. Monotonic models for asymmetric proximities. *Journal of Mathematical Psychology*, 20(1):1–15, 1979.
- [JBG12] Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. Soft cardinality: A parameterized similarity function for text comparison. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 449–453. Association for Computational Linguistics, 2012.
- [JC97] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.
- [JD88] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.
- [JK95] John S Justeson and Slava M Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering*, 1(01):9–27, 1995.
- [JMF99] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.
- [KHK99] George Karypis, Eui-Hong Han, and Vipin Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.
- [Kin82] SF King. Syntactic pattern recognition and applications, 1982.
- [KK98] George Karypis and Vipin Kumar. hmetis 1.5: A hypergraph partitioning package. Technical report, Technical report, Department of Computer Science, University of Minnesota, 1998. Available on the WWW at URL <http://www.cs.umn.edu/metis>, 1998.

- [Knu] Donald Ervin Knuth. *The Stanford GraphBase: a platform for combinatorial computing*, volume 37.
- [LBM03] Yuhua Li, Zuhair A Bandar, and David McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on knowledge and data engineering*, 15(4):871–882, 2003.
- [LC98] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283, 1998.
- [LF09] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.
- [LFR08] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4):046110, 2008.
- [Lin98] Dekang Lin. An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304. Citeseer, 1998.
- [LPCM99] Lillian Lee, Fernando CN Pereira, Claire Cardie, and Raymond Mooney. Similarity-based models of word cooccurrence probabilities. In *Machine Learning*. Citeseer, 1999.
- [Lus03] David Lusseau. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London B: Biological Sciences*, 270(Suppl 2):S186–S188, 2003.
- [LW67] Godfrey N Lance and William T Williams. Mixed-data classificatory programs i - agglomerative systems. *Australian Computer Journal*, 1(1):15–20, 1967.
- [LW71] Francois Lorrain and Harrison C White. Structural equivalence of individuals in social networks. *The Journal of mathematical sociology*, 1(1):49–80, 1971.
- [Mar96] John I Marden. *Analyzing and modeling rank data*. CRC Press, 1996.
- [MCJ06] David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 337–344. Association for Computational Linguistics, 2006.
- [New06] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.
- [NG04] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
- [Nos91] Robert M Nosofsky. Stimulus bias, asymmetric similarity, and classification. *Cognitive Psychology*, 23(1):94–140, 1991.

- [Pan03] Patrick Andre Pantel. *Clustering by committee*. PhD thesis, Citeseer, 2003.
- [PL05] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.
- [PLS04] Witold Pedrycz, Vincenzo Loia, and Sabrina Senatore. P-fcm: a proximity—based fuzzy clustering. *Fuzzy Sets and Systems*, 148(1):21–41, 2004.
- [RCP00] Richard P Runyon, Kay A Coleman, and David J Pittenger. *Fundamentals of behavioral statistics*. McGraw-Hill, 2000.
- [Res] P Resik. Using information content to evaluate semantic similarity. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 448–453.
- [RMBB89] Roy Rada, Hafedh Mili, Ellen Bicknell, and Maria Blettner. Development and application of a metric on semantic nets. *IEEE transactions on systems, man, and cybernetics*, 19(1):17–30, 1989.
- [Sah06] Magnus Sahlgren. The word-space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. 2006.
- [SB08] Sriparna Saha and Sanghamitra Bandyopadhyay. Application of a new symmetry-based cluster validity index for satellite image segmentation. *IEEE Geoscience and Remote Sensing Letters*, 5(2):166–170, 2008.
- [SB13] Sriparna Saha and Sanghamitra Bandyopadhyay. A generalized automatic clustering algorithm in a multiobjective framework. *Applied Soft Computing*, 13(1):89–108, 2013.
- [SC01] Mu-Chun Su and Chien-Hsing Chou. A modified version of the k-means algorithm with a distance based on cluster symmetry. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):674–680, 2001.
- [SEKX98] Jörg Sander, Martin Ester, Hans-Peter Kriegel, and Xiaowei Xu. Density-based clustering in spatial databases: The algorithm gdbSCAN and its applications. *Data mining and knowledge discovery*, 2(2):169–194, 1998.
- [She74] Roger N Shepard. Representation of structure in similarity data: Problems and prospects. *Psychometrika*, 39(4):373–421, 1974.
- [SI84] Shokri Z Selim and Mohamed A Ismail. K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Transactions on pattern analysis and machine intelligence*, (1):81–87, 1984.
- [Spä80] Helmuth Späth. *Cluster analysis algorithms for data reduction and classification of objects*, volume 4. Halsted Pr, 1980.
- [STC15a] J. Santisteban and J.L. Tejada Carcamo. Paradigmatic clustering for nlp. IEEE International Conference on Data Mining Workshops, ICDMW, 2015.

- [STC15b] J. Santisteban and J.L. Tejada Carcamo. Unilateral jaccard similarity coefficient. volume 1393, pages 23–27. CEUR Workshop Proceedings, 2015.
- [STC<sup>+</sup>15c] Julio Santisteban, Javier Tejada-C, et al. Paradigmatic clustering for nlp. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 814–820. IEEE, 2015.
- [Tan05] P Tan. S. m, and kv introduction to data mining, 2005.
- [Tek06] Kardi Teknomo. Similarity measurement. Available: [http://people.revoledu.com/kardi/tutorial/Similarity/\[Accessed\]](http://people.revoledu.com/kardi/tutorial/Similarity/[Accessed]), 2006.
- [TG74] Julius T Tou and Rafael C Gonzalez. Pattern recognition principles. 1974.
- [TK06] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition, Third Edition*. Academic Press, Inc., Orlando, FL, USA, 2006.
- [Tve77a] A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.
- [Tve77b] Amos Tversky. Features of Similarity. In *Psychological Review*, volume 84, pages 327–352, 1977.
- [VL07] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
- [VVR<sup>+</sup>05] Giannis Varelas, Epimenidis Voutsakis, Paraskevi Raftopoulou, Euripides GM Petrakis, and Evangelos E Milios. Semantic similarity methods in wordnet and their application to information retrieval on the web. In *Proceedings of the 7th annual ACM international workshop on Web information and data management*, pages 10–16. ACM, 2005.
- [Web03] Andrew R Webb. *Statistical pattern recognition*. John Wiley & Sons, 2003.
- [WP94] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.
- [WR83] Douglas R White and Karl P Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5(2):193–234, 1983.
- [WW05] Julie Weeds and David Weir. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–475, 2005.
- [WYM<sup>+</sup>97] Wei Wang, Jiong Yang, Richard Muntz, et al. Sting: A statistical information grid approach to spatial data mining. In *VLDB*, volume 97, pages 186–195, 1997.
- [WYM99] Wei Wang, Jiong Yang, and Richard Muntz. Sting+: An approach to active spatial data mining. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 116–125. IEEE, 1999.

- [YL15] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.
- [Zac77] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

# **Apendice A**

## **Artículo I**

---

*Santisteban, J., Tejada Cárcamo, J. (2015). "Unilateral Jaccard Similarity Coefficient". SIGIR (Special Interest Group in Information Retrieval) Workshop on Graph Search and Beyond.*

# Unilateral Jaccard Similarity Coefficient

Julio Santisteban

Universidad Católica San Pablo

Campus Campiña Paisajista s/n Quinta Vivanco,  
Barrio de San Lázaro  
Arequipa, Peru  
jsantisteban@ucsp.edu.pe

Javier L. Tejada Carcamo

Universidad Católica San Pablo

Campus Campiña Paisajista s/n Quinta Vivanco,  
Barrio de San Lázaro  
Arequipa, Peru  
jtejadac@ucsp.edu.pe

## ABSTRACT

Similarity measures are essential to solve many pattern recognition problems such as classification, clustering, and retrieval problems. Various similarity measures are categorized in both syntactic and semantic relationships. In this paper we present a novel similarity, Unilateral Jaccard Similarity Coefficient (uJaccard), which doesn't only take into consideration the space among two points but also the semantics among them.

## Categories and Subject Descriptors

E.1 [Data Structures]: Graphs and networks; G.2.2 [Graph Theory]: Graph algorithms

## General Terms

Theory

## Keywords

Jaccard, distance, similarity

## 1. INTRODUCTION

Since Euclid to today many similarity measures have been developed to consider many scenarios in different areas, particularly in the last century. Similarity measures are used to compare different kind of data which is fundamentally important for pattern classification, clustering, and information retrieval problems [3]. Similarity relations have generally been dominated by geometric models in which objects are represented by points in a Euclidean space [12]. Similarity is defined as "Having the same or nearly the same characteristics" [4], while the metric distance is defined as "The property created by the space between two objects or points". All metric distance functions must satisfy three basic axioms: minimality and equal self-similarity, symmetry, and triangle inequality.

$$d(i, i) = d(j, j) \leq d(i, j) \quad (1)$$

$$d(i, j) = d(j, i) \quad (2)$$

$$d(i, j) + d(j, k) \geq d(i, k) \quad (3)$$

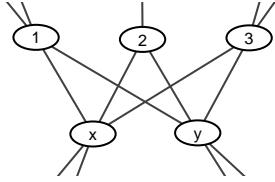
Here for objects i, j and k, where  $d()$  is the distance between objects i and j. Bridge [1] argues that there exists empirical evidence of violations against each of the three axioms. Yet, there also exists geometric models of similarity which take asymmetry into account [10]. Nosofsky points out that a number of well-known models for asymmetric proximity data are closely related to the additive similarity and bias model [5]. Tversky [13] has proposed a different model in order to overcome the metric assumption of geometric models. One of the strengths of contrast models is its capability to explain asymmetric similarity judgments. Tversky's asymmetry may often be characterized in terms of stimulus bias and determined by the relative prominence of the stimuli.

$$\text{sim}(a, b) = \frac{|A \cap B|}{|A \cap B| + \alpha|A - B| + \beta|B - A|}, \quad (4)$$
$$\alpha, \beta \geq 0$$

Here A and B represent feature sets for the objects a and b respectively; the term in the numerator is a function of the set of shared features, a measure of similarity, and the last two terms in the denominator measure dissimilarity:  $\alpha$  and  $\beta$  are real-number weights; when  $\alpha \neq \beta$ . Jimenez et al. [6], Weeds and Weir [14] and Lee [7] also propose an asymmetric similarity measure based on Tversky's work. However all proposals include a stimulus bias, asymmetric similarity judgments, which Tversky refers to as human judgment. Today, similarity measure is deeply embedded into many of the algorithms used for graph classification, clustering and other tasks. Those techniques are leaving aside the semantic of each vertex and it's relation among other vertices and edges.

In a direct graph, the similarity from U to Z is not the same as the distance from Z to U, this due to the intrinsic features of a direct graph. The similarities are different because the channels are dissimilar. According to Shannon's information theory we could argue that each vertex is a source of energy with an average entropy which is shared among its channels, and while that information flow among the vertex's channels, we need to consider it in the similarity. A similarity does not fit all tasks or cases.

In Natural Language Processing, where the similarity between two words is not symmetric  $\text{sim}(\text{word } a, \text{word } b) \neq \text{sim}(\text{word } b, \text{word } a)$ . WordNet [4] presents 28 different types of relations; those relations have direction but are not symmetrical, they are not even synonyms because each synonym word has



**Figure 1: Structural Equivalence.**

a particular semantic, meaning and usage, but are similar. Hence if two words have symmetric distance or similarity, those two words are the same. Paradigmatic is an intrinsic feature in language, It lets the utterer exchange words with other words, words with similar semantics [11]. In this paper we focus on paradigmatic analysis to support our unilateral Jaccard Similarity coefficient (uJaccard).

The rest of the paper is organized as follows. In section 2 we will show the unilateral Jaccard Similarity coefficient (uJaccard). In section 3 we will consider some cases; finally in section 4 we conclude this work.

## 2. PARADIGMATIC SIMILARITY DEFINITION

### 2.1 Basics Of Paradigmatic Structures

Paradigmatic analysis is a process that identifies entities which are not related directly but are related by their properties, relatedness among other entities and interchangeability [2]. In language the reason why we tend to use morphologically unrelated forms in comparative oppositions is to emphasize the semantics, this is done by substitution and transposition of words with a similar signifier. Similarity is not defined by a syntactic set of rules but rather by the use of the language. In some cases this use is not grammatically or syntactically correct but it is commonly used. We defined the signifier as being the degree of relation among entities of the same group, where not all members of the group have the same degree of relatedness. This is due to the fact that a member of a group might belong to more than one group.

### 2.2 Extended Paradigmatic

Two vertices in a graph are structurally equivalent if they share many of the same network neighbours. Figure 1 depicts a structural equivalence between two vertices  $y$  and  $x$  who have the same neighbours. Regular equivalence is more subtle, two regularly equivalent vertices do not necessarily share the same neighbours, but they do have neighbours who are themselves similar [8] [15]. We will use structural equivalence as the bases of uJaccard.

#### 2.2.1 Unilateral Jaccard Similarity

To calculate a paradigmatic similarity we start with a question, is the similarity coefficient from vertex  $V_a$  to  $V_c$  the same to the similarity coefficient from vertex  $V_c$  to  $V_a$ ? If we argue that both similarity coefficients are the same, we are arguing that the edges from the vertices  $V_a$  and  $V_c$  are the same, and it is clear that that is not usually the case. Thus both vertices have different sets of edges. One problem with Tversky [13] similarity is the estimation for  $\alpha$  and  $\beta$  which are stimulus bias, generally a human factor. Similarly, other similarities which are based on Tversky idea, have the

same problem. On the other hand we propose a measure that does not include this bias. We propose a modified version of Jaccard Similarity coefficient (1), unilateral Jaccard Similarity coefficient (uJaccard) (2)(3), used to identify the similarity coefficient of  $V_a$  to  $V_c$  With respect to vertex  $V_a$ , and to also identify the similarity coefficient of  $V_c$  to  $V_a$  With respect to vertex  $V_c$ .

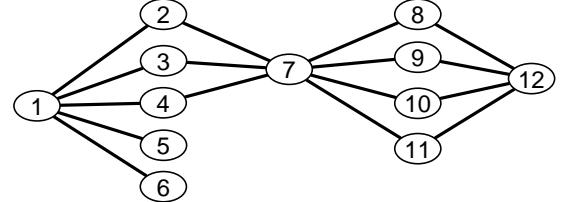
$$Jaccard(V_a, V_c) = \frac{|a \cap c|}{|a \cup c|} \quad (5)$$

$$uJaccard(V_a, V_c) = \frac{|a \cap c|}{|edges(a)|} \quad (6)$$

$$uJaccard(V_c, V_a) = \frac{|c \cap a|}{|edges(c)|} \quad (7)$$

Here  $V_a$  and  $V_c$  are the number of edges in vertex  $a$  and  $c$ , likewise the  $edges(V_c)$  are the number of edges in vertex  $c$ . if  $uJaccard$  is close to 0, it means that they are not similar at all. The objective of using  $uJaccard$  is to identify how similar a vertex is to other vertices in relation to itself.  $uJaccard$  could be calculated among two connected vertices,  $uJaccard$  could also be calculated among vertices that are not connected directly, but which are connected by in-between vertices. The number of in-between vertices could be from 1 to  $n$ , we do not recommend a deep comparison since the semantics of the vertex loosest its meaning. Hence  $\max(n)=3$ , it is suggested for NLP. For the calculations we do not consider the number of in-between vertices since we focus on the information flow and not the information transformation carried out on the intermediate vertices.

## 3. EXPERIMENTAL EVALUATION



**Figure 2: Toy graph.**

### 3.1 Toy Testing

Using similarity  $uJaccard$  (6),(7) we can build a paradigmatic approach to group vertices. Figure 2 shows a toy graph with 12 vertices and 16 edges, following the paradigmatic analysis, we can determine that vertex 12 and 7 belong to group P because they have the same number of edges to a same set of vertices. Vertex 1 also belongs to group P because vertex 1 has 3 of the 5 edges, the same as vertex 7, the degree of membership of vertex 1 is lower than vertices 7 and 12 because vertex 1 has other edges that are not shared by vertices 7 or 12. In the same manner we can determine that vertex 8, 9, 10 and 11 belong to group Q because they have an equal number of edges to the same set of vertices. Similarly vertices 2, 3 and 4 belong to group R, and vertices 5 and 6 belong to group O. In this example we can easily identify the paradigmatic approach, where two or more

vertices belong to the same group if they have the same or similar neighbours, but the neighbours in turn belong to another group.

Following the uJaccard similarity and the paradigmatic

**Table 1: uJaccard calculation from figure 3**

uJaccard (V1,V7) = 3/5 = 0.600
uJaccard (V7,V1) = 3/7 = 0.428
uJaccard (V7,V12) = 4/7 = 0.571
uJaccard (V12,V7) = 4/4 = 1.000
Jaccard (V1,V7) = 3/9 = 0.333
Jaccard (V7,V1) = 3/9 = 0.333
Jaccard (V7,V12) = 4/7 = 0.571
Jaccard (V12,V7) = 4/7 = 0.571

approach, the results of the graph in figure 3 are shown in table 3.1. we notice that uJaccard similarity provides better information of similarity than Jaccard, this is because uJaccard considers the notion of unilateral similarity. Table 3.1 shows three toy graphs, in which we present a comparison between Jaccard and uJaccard. As show in table 3.1 uJaccard provides a unilateral similarity improving the symmetric similarity Jaccard.

Table 3.1 shows three toy graphs, in which we present a

**Table 2: Test uJaccard in toy graphs**

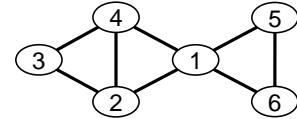
	uJaccard sim(2,5)=4/4 sim(5,2)=4/6
	uJaccard sim(7,5)=1/3 sim(5,7)=1/1
	uJaccard sim(2,4)=3/4 sim(4,2)=3/5
	Jaccard sim(2,5)=4/6 sim(5,2)=4/6
	Jaccard sim(7,5)=1/3 sim(5,7)=1/3
	Jaccard sim(2,4)=3/6 sim(4,2)=3/6

comparison among Jaccard and uJaccard. As show in table 3.1 uJaccard provide an unilateral similarity improving the symmetric similarity Jaccard.

### 3.2 Cut a graph

In graph theory, a cut is a partition of the vertices of a graph into two disjoint subsets. There are many techniques and algorithms to cut a graph, but in some cases there are graphs that are difficult to cut, due to their symmetric distribution of vertices.

It is shown in figure 3.2 that node 1 might belong to cluster {2,3,4} or cluster {5,6}; to resolve this problem we use uJaccard similarity measure to find the similarity of node 1 to other nodes. Table 3 shows that similarities from node 1 to other nodes 1 level deep are the same, so we could not allocate node 1 to a particular cluster. Table 3 also shows that similarities from node 1 to other nodes 2 levels deep, in which uJaccard(1,3) has a strong similarity over the rest. We could conclude that node 1 belong to cluster {2,3,4}. In figure 3.2 also node 1 might belong to cluster {2,3,4} or

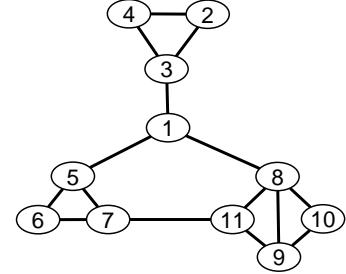


**Figure 3: Toy graph.**

**Table 3: Cut a graph 3.2 using uJaccard**

1 level deep	2 levels deep
uJaccard(1,4)	1/4
uJaccard(1,2)	1/4
uJaccard(1,3)	2/4
uJaccard(1,5)	1/4
uJaccard(1,4)	1/4
uJaccard(1,6)	1/4
uJaccard(1,5)	1/4
-	-
uJaccard(1,6)	1/4

cluster {5,6,7} or cluster {8,9,10,11}; this is where uJaccard comes in, being able to solve this problem. Table 4 shows result of similarities from node 1 to all other nodes on the network in different levels deep. cluster {8,9,10,11} presents the highest number of strong similarities, therefor we can conclude that node 1 belongs to cluster {8,9,10,11}.



**Figure 4: Toy graph.**

### 3.3 Social Network

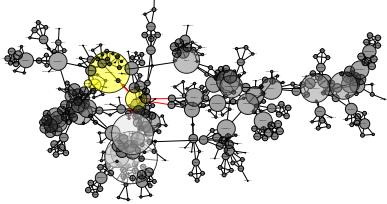
We tested uJaccard against two social network graphs; the first is the coauthorship network of scientists [9] the second is the network of Hollywood's actors<sup>1</sup>.

The first network is the coauthorship network of scientists working on network theory and experiments, compiled by M. Newman [9]. We want to find the top scientists that Newman is similar to or that have paradigmatic similarity. As shown in table 5 the 3 most of Newman's paradigmatic similar scientists are Callaway, Strogatz and Holme. On the

<sup>1</sup>The Internet Movie Database: <ftp://ftp.fu-berlin.de/pub/misc/movies/database/>

**Table 4: Cut a graph 3.2 using uJaccard**

2 levels deep	3 levels deep
uJaccard(1,4)	1/3
uJaccard(1,2)	1/3
uJaccard(1,6)	1/3
uJaccard(1,7)	1/3
uJaccard(1,9)	1/3
uJaccard(1,11)	1/3
uJaccard(1,10)	1/3



**Figure 5: Coauthorship network of scientists, selected nodes belong to scientists Newman, Callaway, Strogatz, Holme.**

other hand the top 3 scientists that are similar to Newman are Adler, Aberg and Aharony. uJaccard has been calculated in 2,3 and 4 levels deep away from Newman. Newman is more similar to Strogatz but the most similar scientist to new Newman is Adler and not Strogatz, even that Strogatz most similarity is toward Newman.

For the second network, we created the second social network of Hollywood's actors, we based on The Internet Movie Database (note). We download actors and actresses data, which includes title of movies in which they worked, we also download a list of top 1000 (nota) and top 250 (nota) actors and actresses. The network is composed of nodes representing actors and actresses, and vertices are the movies in which those actors worked together. A node is created for every person, with their names as the key, when two people are in the same movie; a vertex is created between their nodes. The first network presents 1000 top actors and actresses who also work in 41,719 movies with a total 113,478 edges. The second network presents 250 actors and actresses who work in 15,831 movies with a total of 14,096 edges. For this test we remove duplicated edges.

- From a given *actor A*
- We search for actors that *actor A* is similar to
- From the *actor A*'s similar actor list we get the most similar *actor B*
- We search for actors that *actor B* is similar to
- This is done to analyse if *actor A* and *actor B* are reciprocally similar
- Then we look for actors that are most similar to *actor A*
- We do this on the network top 250 actors and top 1000 actors.

**Table 5: uJaccard calculation from 3.3, in search paradigmatic scientists to Newman**

2 levels deep	3 levels deep	4 levels deep
Scientist Newman is similar to:		
Callaway	0.15	Strogatz
Strogatz	0.15	Holme
Watts	0.15	Kleinberg
Hopcroft	0.11	Sole
Scientists that are similar to Newman:		
Adler	0.33	Aberg
Aharony	0.33	Adler
Aleksiejuk	0.50	Aharony
Ancelmeyers	0.66	Alava
Araujo	0.33	Albert

The results of the search on the network of top 250 actors and top 1000 actors, using uJaccard and the paradigmatic approach are presented in tables 6 and 7. In table 6 we focus in *Tom Cruise*, we found that *Tom Cruise* is most similar to *Julia Roberts* but *Julia Roberts* is most similar to *John Travolta*, *Tom Cruise* is third in *Julia Roberts'* similarity list. clearly there is not a symmetric similarity among *Julia Roberts* and *Tom Cruise*. Moreover *Julia Roberts* is not the most similar toward *Tom Cruise*, the most similar towards *Tom Cruise* is *Heath Ledger*. Hence this confirm that uJaccard helps to identify similarities, particularly asymmetric similarities. Table 6 also shows similar scenario among *Tom Cruise*, *Tom Hanks* and *Joan Allen* in the network of top 250 actors and actresses, this confirm the usability of uJaccard.

In Table 7 we use the network of top 250 actors and ac-

**Table 6: uJaccard similarity among top 1000 and top 250 actor and actresses, searching paradigmatic similar actor**

Top 1000 actors, cruise tom is similar to:	
roberts julia	0.405
hanks tom	0.401
jackson samuel	0.399
douglas michael	0.397
eastwood clint	0.393
Top 250 actors, cruise tom is similar to:	
hanks tom	0.430
douglas michael	0.420
eastwood clint	0.420
spacey kevin	0.413
jackson samuel	0.410
Who is similar to cruise tom:	
top 1000 actors	
ledger heath	0.490
bacon kevin	0.488
crowe russell	0.482
gibson mel	0.482
benigni roberto	0.482
top 250 actors	
allen joan	0.496
balk fairuza	0.495
bello maria	0.488
collins pauline	0.487
aiello danny	0.487

tresses and we focus on *Anthony Quinn* and *Jack Nicholson*. We start by searching for actors that *Anthony Quinn* is sim-

ilar to, then we search for actors that are most similar to *Anthony Quinn* in 1 and 2 levels deep. We notice that *Anthony Quinn* is most similar to *Tom Hanks* but most similar actor to *Anthony Quinn* is *Antonio Banderas*, while *Anthony Quinn* is the 153th most similar for *Tom Hanks*. *Antonio Banderas* is most similar to *Samuel Jackson* and not to *Anthony Quinn*, while *Anthony Quinn* is the 53th most similar for *Antonio Banderas*. Therefore we could conclude that *Anthony Quinn* and *Tom Hanks* are not symmetric similar rather they are asymmetric similar. Table 7 also shows similar scenario for *Jack Nicholson*.

**Table 7: uJaccard similarity among top 250 actor and actresses, searching paradigmatic actor, in 2 and 3 levels deep**

Top 250 actors, are similar to:			
quinn anthony		nicholson jack	
hanks tom	0.451	hanks tom	0.436
jackson samuel	0.443	eastwood clint	0.429
lemon jack	0.443	travolta john	0.417
cruise tom	0.435	williams robin	0.417
de niro robert	0.435	douglas michael	0.414
Who are similar to quinn anthony:			
1 level deep		2 levels deep	
banderas antonio	0.366	banderas antonio	21.866
bardem javier	0.350	benigni roberto	20.758
martin steve	0.333	burns george	20.565
goodman john	0.320	baldwin alec	20.413
allen woody	0.285	mcqueen steve	20.244
Who are similar to nicholson jack:			
1 level deep		2 levels deep	
greene graham	0.484	banderas antonio	41.477
bronson charles	0.482	bacon kevin	41.133
brody adrien	0.480	baldwin alec	41.0862
bale christian	0.476	bronson charles	40.982
baldwin alec	0.474	benigni roberto	40.948

## 4. CONCLUSION

A key assumption of most models of similarity is that a similarity relation is symmetric. The symmetry assumption is not universal, and it is not essential to all applications of similarity. The need for asymmetric similarity is important and central in Information Retrieval and Graph Data Networks. It can improve current methods and provide an alternative point of view.

We present a novel asymmetric similarity, Unilateral Jaccard Similarity (uJaccard), where the similarity among A and B is not same to the similarity among B and C,  $uJaccard(A,B) \neq uJaccard(B,A)$ ; this is based on the idea of paradigmatic association. In comparison to Tversky [13] our approach uJaccard does not need a stimulus bias, whereas in the case of Tversky human judgement is needed.

We present a series of cases in which we confirmed its usefulness and we validated uJaccard. We could extend uJaccard to include weights to improve the asymmetry, we could also use uJaccard and the paradigmatic approach to cluster Graph data Networks. These are tasks in which we are working on.

In conclusion, the proposed uJaccard similarity proved to be useful despite its simplicity and the few resources used.

## 5. ACKNOWLEDGMENTS

This research was funded in part by "Fondo para la Innovación, Ciencia y Tecnología - Peru" (FINCyT). We thank the Universidad Católica San Pablo for supporting this research.

## 6. REFERENCES

- [1] D. Bridge. Defining and combining symmetric and asymmetric similarity measures. In B. Smyth and P. Cunningham, editors, *Advances in Case-Based Reasoning (Procs. of the 4th European Workshop on Case-Based Reasoning)*, LNAI 1488, pages 52–63. Springer, 1998.
- [2] F. De Saussure and W. Baskin. *Course in general linguistics*. Columbia University Press, 2011.
- [3] R. Duda, P. Hart, and D. Stork. Pattern classification 2nd ed., 2001.
- [4] C. Fellbaum. Wordnet and wordnets. In A. Barber, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier, 2005.
- [5] E. W. Holman. Monotonic models for asymmetric proximities. *Journal of Mathematical Psychology*, 20(1):1–15, 1979.
- [6] S. Jimenez, C. Becerra, and A. Gelbukh. Soft cardinality: A parameterized similarity function for text comparison. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 449–453. Association for Computational Linguistics, 2012.
- [7] L. Lee, F. C. Pereira, C. Cardie, and R. Mooney. Similarity-based models of word cooccurrence probabilities. In *Machine Learning*. Citeseer, 1999.
- [8] F. Lorrain and H. C. White. Structural equivalence of individuals in social networks. *The Journal of mathematical sociology*, 1(1):49–80, 1971.
- [9] M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.
- [10] R. M. Nosofsky. Stimulus bias, asymmetric similarity, and classification. *Cognitive Psychology*, 23(1):94–140, 1991.
- [11] M. Sahlgren. The word-space model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces. 2006.
- [12] R. N. Shepard. Representation of structure in similarity data: Problems and prospects. *Psychometrika*, 39(4):373–421, 1974.
- [13] A. Tversky. Features of Similarity. In *Psychological Review*, volume 84, pages 327–352, 1977.
- [14] J. Weeds and D. Weir. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–475, 2005.
- [15] D. R. White and K. P. Reitz. Graph and semigroup homomorphisms on networks of relations. *Social Networks*, 5(2):193–234, 1983.

# **Apendice B**

## **Artículo II**

---

*Santisteban, J., Tejada Cárcamo, J. (October-2015). "Unilateral Weighted Jaccard Coefficient for NLP". IEEE Mexican International Conference in Artificial Conference.*

## Unilateral Weighted Jaccard Coefficient for NLP

Julio Santisteban

Universidad Católica San Pablo  
Campus Campiña Paisajista s/n  
Quinta Vivanco, Barrio de San Lázaro  
Arequipa, Peru  
[jsantisteban@ucsp.edu.pe](mailto:jsantisteban@ucsp.edu.pe)

Javier Tejada-Cárcamo

Universidad Católica San Pablo  
Campus Campiña Paisajista s/n  
Quinta Vivanco, Barrio de San Lázaro  
Arequipa, Peru  
[jtejadac@ucsp.edu.pe](mailto:jtejadac@ucsp.edu.pe)

**Abstract**—Similarity measures are essential to solve many pattern recognition problems such as classification, clustering, and retrieval problems. Various similarity measures are categorized in both syntactic and semantic relationships. In this paper we present a novel similarity, Unilateral Weighted Jaccard Coefficient (uwJaccard), which takes into consideration not only the space among two points but also the semantics among them in a distributional semantic model, the Unilateral Weighted Jaccard Coefficient provides a measure of uncertainty which will be able to measure the uncertainty among sentences such as “*man bites dog*” and “*dog bites man*”.

**Keywords**-Jaccard; distance; similarity; entropy; uncertainty;

### I. INTRODUCTION

Since Euclid to today many similarity measures have been developed to be used under many scenarios and in different areas, particularly in the last century. Similarity measures are used to compare different kinds of data which is fundamentally important for pattern classification, clustering, and information retrieval problems [1]. Similarity relations have generally been dominated by geometric models in which objects are represented by points in a Euclidean space [2]. Similarity is defined as “Having the same or nearly the same characteristics” [3], while the metric distance is defined as “The property created by the space between two objects or points”. All metric distance functions must satisfy three basic axioms: minimality and equal self-similarity, symmetry, and triangle inequality [4]. Bridge [5] argues that there exists empirical evidence of violations against each of the three axioms. Yet, there also exists geometric models of similarity which take asymmetry into account [6]. Nosofsky points out that a number of well-known models for asymmetric proximity data are closely related to the additive similarity and bias model [7]. Tversky [8] has proposed a different model in order to overcome the metric assumption of geometric models. One of the strengths of contrast models is its capability to explain asymmetric similarity judgments. Tversky’s asymmetry may often be characterized in terms of stimulus bias and determined by the relative prominence of the stimuli. Jimenez et al. [9], Weeds and Weir [1] and Lee [10]

also propose an asymmetric similarity measure based on Tversky’s work. However all proposals include a stimulus bias, asymmetric similarity judgments, which Tversky refers to as human judgment.

Today, similarity measure is deeply embedded into many of the algorithms used for graph classification, clustering and other tasks. Those techniques left aside the semantic of each vertex and it’s relation among other vertices and edges [11],[12]. In a graph classification and clustering we consider only the features of a vertex, edge and its cumulus, from those basic measurements many techniques have been developed for classification and clustering with very good results. Most of those techniques explore a symmetric distance or similarity among two points, leaving aside the semantic of each vertex and it’s relation among other vertices and edges.

In a direct graph, the distance from  $U$  to  $Z$  is not the same as the distance from  $Z$  to  $U$ , it is because the intrinsic features of a direct graph. The distance are different because the channels are dissimilar. According to Shannon’s information theory we could argue that the each vertex is a source of energy with an average entropy which is shared among it’s edges, hence that information flows among the vertex’s edges therefore it needs to be consider in the distance or similarity coefficient. The entropy been its meaning and semantics. We will use the term similarity since we look at vertex’s semantics and not just its location in the space. A similarity does not fit all tasks or cases. In this paper we try to focus on Natural Langue Processing, where the similarity between two word is not symmetric  $\text{sim}(\text{word } A, \text{word } B) \neq \text{sim}(\text{word } B, \text{word } A)$ . WordNet [13] present 28 different types of relations, those relations have direction and are not symmetrical, they are not even synonyms because each synonym word has a particular semantic, meaning and usage, but they are similar. Hence if two words have symmetric distance or similarity, those two words are the same.

The rest of the paper is organized as follows. In section 2 we will review our previous work Unilateral Jaccard Coefficient Similarity coefficient (uJaccard). In section 3 we will present Unilateral Weighted Jaccard Coefficient and we will test it.

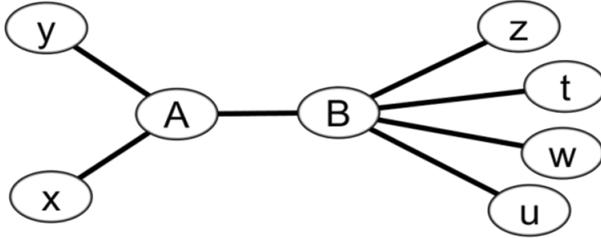


Figure 1: Simple graph

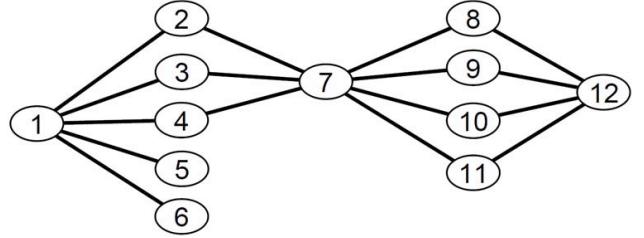


Figure 2: Toy graph

Table I: My caption

uJaccard similarity from toy graph Figure 2	
uJaccard (V1,V7) = 3/5	Jaccard (V1,V7) = 3/9
uJaccard (V7,V1) = 3/7	Jaccard (V7,V1) = 3/9
uJaccard (V7,V12) = 4/7	Jaccard (V7,V12)= 4/7
uJaccard (V12,V7) = 4/4	Jaccard (V12,V7)= 4/7

Finally, we summarize and draw some conclusions in section 4.

## II. UNILATERAL JACCARD COEFFICIENT SIMILARITY

To calculate the similarity we start with one question; is the similarity coefficient from vertex  $V_a$  to  $V_c$  the same as the similarity coefficient from vertex  $V_c$  to  $V_a$ ? If we argue that both similarity coefficients are the same, we are arguing that the edges from the vertices  $V_a$  and  $V_c$  are the same, and it is clear that is not usually the case. Thus both vertices have different sets of edges.

The Unilateral Jaccard Coefficient similarity coefficient is used to find those vertices that have a structural equivalence [14] [15]. Jaccard similarity consider just one coefficient on the contrary Unilateral Jaccard Coefficient provides two coefficients since it is an asymmetric similarity, which are define by:

$$Jaccard(a, c) = \frac{|a \cap c|}{|a \cup c|} \quad (1)$$

$$uJaccard(a, c) = \frac{|a \cap c|}{|edges_a|} \quad (2)$$

$$uJaccard(c, a) = \frac{|c \cap a|}{|edges_c|} \quad (3)$$

Where  $a$  and  $c$  are vertices in a network graph.  $a \cap c$  is the intersection of edges between vertices  $a$  and  $c$ , and  $edges_a$  is the number of edges in vertex  $a$ .

We review the propose modification version of Jaccard Similarity (1), Unilateral Jaccard Coefficient Similarity (uJaccard) (2)(3) [14], which identifies the similarity coefficient of  $V_a$  to  $V_c$  with respect to vertex  $V_a$ , and also identifies the similarity coefficient of  $V_c$  to  $V_a$  with respect to vertex  $V_c$ .

Following Shannon's information theory, we argue that each vertex is a source of energy with an average entropy which is shared among its edges, hence the information flows through the vertex's edges. Figure 1 shows a small graph, the similarity among vertex  $A$  and  $B$  is as follow:  $uJaccard(A,B) = 1/3 = 0.333$ ;  $uJaccard(B,C)=1/5 = 0.2$  and the  $Jaccard(A,B) = 1/7 = 0.143$ ;  $Jaccard(B,A) = 1/7 = 0.143$ . The Jaccard coefficient provides one general measure of similarity among vertex  $A$  and  $B$  however uJaccard provides

a unilateral measure of similarity in which  $A-B$  is different to  $B-C$ .

### A. $N$ levels

uJaccard allows us to measure the similarity among two vertices that are not directly linked like vertices  $A$  and  $B$  in figure 1, but rather indirectly linked in a graph, in which the levels of the indirect link would be  $1$  or  $n$ . like vertices  $1, 7, 12$  in figure 2 which are linked with level  $n = 2$ . The semantic information will start losing its meaning among linked vertices when  $n$  is greater than  $3$ .

We argument that the paths in between the linked vertices are considered in the calculations but the information of each vertex found on that particular path is not considered, this is done to make the model simpler. The focus is the linked vertices not the in-between vertices even if they provide some information for the calculations; we argue that such information is minimal.

The objective of using uJaccard is to identify how similar a vertex is to other vertices in relation to itself. uJaccard is calculated among vertices that are not connected directly, rather are connected by intermedia vertices, of one level. Figure 2 shows a toy graph with  $12$  vertices and  $16$  edges, using Jaccard and uJaccard to calculate its similarity as shown in table I.

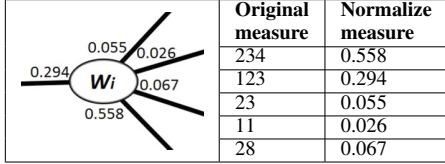
As per the toy example in figure 2, we can notice that Ujaccard similarity provides better information than Jaccard similarity, this is because Ujaccard considers the notion of unilaterally.

Following the results in table I we can conclude that vertices  $1, 12, 7$  are similar. Vertices  $8, 9, 10, 11$  are similar. Vertices  $2, 3, 4$  are similar and vertices  $5, 6$  are also similar. If uJaccard is close to  $1$ , it means that it is  $100\%$  similar, but if it is close to  $0$ , it means that they are not similar at all.

Table II: Comparison uJaccard and Jaccard in toy graph

	uJaccard (7,5) = 1/3 uJaccard (5,7) = 1/1 Jaccard (7,5) = 1/3 Jaccard (5,7) = 1/3
	uJaccard (2,5) = 4/4 uJaccard (5,2) = 4/6 Jaccard (2,5) = 4/6 Jaccard (5,2) = 4/6
	uJaccard (2,4) = 3/4 uJaccard (4,2) = 3/5 Jaccard (2,4) = 3/6 Jaccard (4,2) = 3/6

Table III: Normalized feature vector of a given word  $W_i$



### B. Toy Testing

We evaluated the uJaccard on toy graphs and compared it with its contender Jaccard as shown in table II. where uJaccard provides a more meaningful similarity coefficient than Jaccard.

### III. UNILATERAL WEIGHTED JACCARD COEFFICIENT

Weights for a distributed semantic model(DSM) is an open problem as shown by Grefenstette and Sadrzadeh [6], being the key part for a DSM. It is clear that each constructed weighted model is related to a particular context therefore we could not compare two DSMs. In this article we present a generic process to generate weights for a DSM and we also present a Unilateral Weighted Jaccard Coefficient similarity. Consider a graph of words, where two words are directly linked  $W_1$  linked to  $W_2$ . We will consider the weights as a its frequencies or probabilities or collocation measure such as probabilities  $P(W_1, W_2)$ ,  $P(W_2, W_1)$ ,  $P(W_1, W_n)$  and  $P(W_2, W_n)$ . Consequently edge among vertex  $W_1$  and vertex  $W_2$  will have two measures. Since word  $W_1$  is linked to word  $W_2$  with a particular collocation measure, also word  $W_2$  is linked with word  $W_1$  with a different collocation measure. Hence we end up with a link between word  $W_1$  and word  $W_2$  with two collocations measures. The collocation measures are not the same among red-car that car-red. from now on we will start calling them weights.

A word represented by a vertex in a network graph, the word will have a feature vector which will represent the

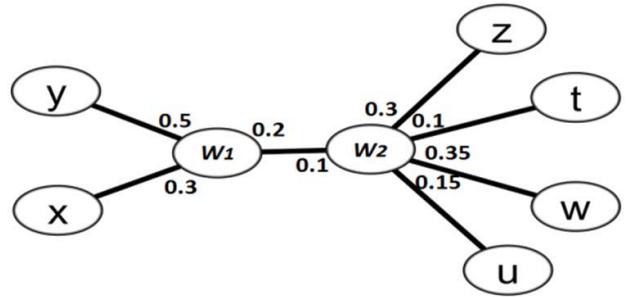


Figure 3: Edge among word  $W_1$  and word  $W_2$  with two weights 0.2 and 0.1 respectively

words that are most similar to it. The feature vector will also have a measure that is only related to the given word and its related words. The feature vector is a large vector and it needs to be truncated in order to get the more meaningful feature words, this can be done by different means. Once the smaller vector has been obtained we normalize the measures, so that all the measures sum 1 as shown in table III. Hence why edge among word  $W_1$  and word  $W_2$  will have two normalize weights as show in figure 3. However each normalized weight is only related to its word and is not related to the linked word. Therefore we conclude that  $P(W_1, W_2)$  and  $P(W_2, W_1)$  belong to two different models and are not related. To solve this problem we use Unilateral Weighted Jaccard Coefficient.

To calculate Unilateral Weighted Jaccard Coefficient we consider two assumptions. First a vertex's information entropy is calculated based on its feature vector weights (4), we will consider the vector's feature weights as the distribution probability for the given word. For a discrete random variable  $X$  that can take values  $x_1, \dots, x_n$  with probability  $p_i$  respectively, defines the entropy  $H(X)$  of  $X$ . Where  $X$  is the word and probability  $p_1, \dots, p_n$  are the vector's feature weights. The information entropy is given by:

$$H(X) = - \sum_{i=1}^n p_i * \log p_i \quad (4)$$

Secondly, we use the cross-entropy (5) to find the similarities  $\text{sim}(W_1, W_2)$  and  $\text{sim}(W_2, W_1)$  as presented by Jurafsky, Martin (2009). The cross-entropy also known as the relative entropy of  $p$  with respect to  $q$  and the relative entropy of  $q$  with respect to  $p$ , as a consequence the cross-entropy is not symmetric. Where  $x$  is a common element in distribution of  $q$  and  $p$ ,  $P(x)$  is the probability of the element  $x$  on distribution  $p$ ,  $Q(x)$  is the probability of the element  $x$  on distribution  $q$ .  $x$  is a common element in distribution  $p$  and  $q$ , as a consequence the calculation for the cross-entropy is done only with the edge(s) linking word  $W_1$  and word  $W_2$ . Hence, the Unilateral Weighted Jaccard Coefficient only considers  $x$  as the links among word  $W_1$  and word  $W_2$ . The

Table IV: The Unilateral Weighted Jaccard Coefficient (uw-Jaccard) among word  $W1$  and word  $W2$

$H(W1) = 1.4855$
$H(W2) = 2.1261$
$H(W1, W2) = 0.6644$
$H(W2, W1) = 0.2322$
$uw\text{Jaccard}(W1, W2) = 2.1499$
$uw\text{Jaccard}(W2, W1) = 2.3583$

cross-entropy is given by:

$$H(P, Q) = - \sum_{i=1}^n P(i) \log Q(i) \quad (5)$$

The Unilateral Weighted Jaccard Coefficient (6),(7) is the sum of the entropy of the vertex  $H(W1)$  and the cross-entropy on the edge  $H(W1, W2)$ . Table 4 shows the *Unilateral Weighted Jaccard Coefficient* ( $W1, W2$ ) and *Unilateral Weighted Jaccard Coefficient* ( $W2, W1$ ) for the edge among vertices word  $W1$  and word  $W2$  from figure 3. The Unilateral Weighted Jaccard Coefficient is given by:

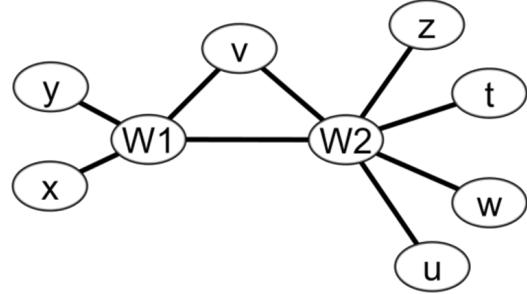
$$uw\text{Jaccard}(p, q) = H(p) + H(p, q) \quad (6)$$

$$uw\text{Jaccard}(q, p) = H(q) + H(q, p) \quad (7)$$

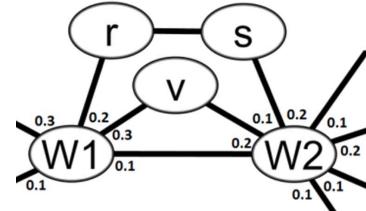
Since we cannot measure two distinct vertices models. We argue that it is superior to observe how uncertain is word  $W1$  to word  $W2$  and how uncertain is word  $W2$  to word  $W1$ . Unilateral Weighted Jaccard Coefficient will measure the uncertainty of a similarity among words  $W1$  and word  $W2$ , in a particular edge. Unilateral Weighted Jaccard Coefficient measure is a positive value, where 0 means perfect similarity and any value greater than 0 means uncertainty.

Thus, the Unilateral Weighted Jaccard Coefficient is not a proper weight rather a measure of information content among words, by no means it is a measure of similarity rather the uncertainty of the link in between the words.

Similarity is not considered only by a direct link but also by an indirect link, in the case of there existing multiple paths from word  $W1$  and word  $W2$ , as shown in figure 4. The Unilateral Weighted Jaccard Coefficient (8) is adjusted to consider multiple paths among vertex  $W1$  and vertex  $W2$ . We might need to consider all possible paths and its segments, which might be a bit complex and cumbersome to carry out, which is why we present a simplified approach. We sum each path to the calculation. For each path from vertex  $W1$  to vertex  $W2$  we calculate the cross-entropy but we only consider the outbound probabilities from vertex  $W1$  and vertex  $W2$ , the cross-entropy is multiplied by  $1 + \lambda$ . Where  $\lambda$  represents the number of vertices on the path, which will increase and decrease the uncertainty in accordance to the number of paths. The The Unilateral Weighted Jaccard



(a) paths without weights



(b) paths with weights

Figure 4: Edge among word  $W1$  and word  $W2$  with multiple paths

Coefficient is given by:

$$uw\text{Jaccard}(p, q) = H(p) - H(pO) \sum_{\text{path } \exists p \leftrightarrow q} (1+\lambda)H(p, q) \quad (8)$$

$$uw\text{Jaccard}(q, p) = H(q) - H(qO) \sum_{\text{path } \exists q \leftrightarrow p} (1+\lambda)H(q, p) \quad (9)$$

Where  $\lambda = 0.1$  times the number of in-between vertices in a path,  $H(p, q)$  and  $H(q, p)$  is the cross-entropy of a given path,  $H(p)$  and  $H(q)$  is the entropy of the originating vertex.  $H(pO)$  and  $H(qO)$  are the entropy of the outbound links from the originating vertex. The path  $\exists q \leftrightarrow p$  are all possible paths that exist between  $q$  and  $p$ , the path can be 1 or  $n$ , but for NLP tasks we shouldn't go further than 3, because we start losing its meaning.

$\lambda$  will increase the uncertainty, the more vertices there are on the path the more the meaning of the similarity decreases, while if there are more number of paths, the similarity increases, therefore reducing its uncertainty as the word is more couple.

The number of paths might introduce the concept of coupling, as the number of connections increase, but it also introduces the uncertainty. There is less uncertainty when the link is direct. As the number of paths increase and the number of vertices on the paths increase the uncertainty also increases. This mean that the similarity decreases even more when we introduce more paths with in-between vertices.

Table V: The Unilateral Weighted Jaccard Coefficient (uw-Jaccard) among word  $W_1$  and word  $W_2$ , with different paths, from graph in figure 4b

H(W1) = 2.170950594
H(W2) = 2.721928095
H(W1 outbound) = 1.317668107
H(W2 outbound) = 1.260964047
H(W1-R-S-W2) = 0.557262743
H(W1-V-W2) = 1.096236271
H(W1-W2) = 0.232192809
H(W2-R-S-W1) = 0.557262743
H(W2-V-W1) = 0.191066215
H(W2-W1) = 0.664385619
<b>Including 3 paths</b>
uwJaccard(W1,W2) = 3.661341986
uwJaccard(W2,W1) = 3.756353458
<b>Including 2 paths</b>
uwJaccard(W1,W2) = 3.367612865
uwJaccard(W2,W1) = 3.451283524
<b>Including a direct paths</b>
uwJaccard(W1,W2) = 2.403143404
uwJaccard(W2,W1) = 3.386313714

Table VI: Size of the used corpus

Corpus	Google bi-gram	BNC	Wikipedia
Size	1.5GB	539MB	11GB
bi-grams	314,843,401	842,162,318	
Words	629,298,853	98,537,224	1,459,026,075
Vocabulary	5,827,772	354,374	3,990,248

#### A. Building a network graph of words

We start with a matrix of words and then we build a graph of words. To build a matrix of words we source it from three main corpus, Google n-gram corpus [16], The British National Corpus, version 3, BNC XML Edition [17] and en.wikipedia database backup dumps.

We start building a bi-grams model from each of the corpus. The construction of bi-grams was straight forward for Google n-gram as it was in fact a bi-gram dataset [16]. With the en.wikipedia and the BNC corpus we had to run a sentence segmentation script against them in order to extract sentences. Once the sentences were identified we parse them to discover POS (part of speech), we use the BLLIP parser at T50 [1]. Once the parsed sentences are obtained we remove the stop words and we apply a lemmatization process; then we start building bi-grams with the following forms, verb-verb, noun-noun, verb-noun, noun-verb, adverb-verb, verb-adverb, adjective-noun, noun-adjective. We use a window of  $n$  to the left and/or right of a the working word to build the bi-grams; after a few experiments we use  $n = 3$ , a window larger than 3 produces meaningless semantics, this being an extension work of Justeson and Katz [18] and Patrick Pantel. [19].

The bi-grams frequency provides a generic collocation measure, to improve the model we select two collocation measures from the state of the art, Log-likelihood Ratios

(LLR) and we assume a binomial distribution for a word in the text. LLR has been selected because it is a good approach to sparse data and it is able to identify rare bi-grams with a low frequency. The Log-likelihood Ratios is given by:

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)} \quad (10)$$

$$\log \lambda = \log \frac{b(c_{12}, c_1, p) b(c_2 - c_{12}, N - c_1, p)}{b(c_{12}, c_1, p_1) b(c_2 - c_{12}, N - c_1, p_2)} \quad (11)$$

The second collocation measure we selected is the Pointwise Mutual Information between two events. In information theory, mutual information refers to the mutual information between two random variables rather than between two related events. The Log-likelihood Ratios is given by:

$$PMI(x, y) = \log \frac{p(x, y)}{p(x)p(y)} \quad (12)$$

PMI is bad with sparse data, some words that only occur once, but appear together, score very high PMI. PMI is also bad with word dependence, when two words are perfectly dependent of each other, whenever one occurs, the other occurs, so the rarer the word is, the higher the PMI it gets. In these two cases a higher PMI doesn't necessarily indicate high collocation. Bouma, G. [20] presents several normalized variants of PMI in order to make PMI less sensitive to low frequency. For large sparse data sets like ours the Normalized Pointwise Mutual Information to the power of cube  $NPMI^3$  is the best approach, as those low frequencies do not surface easily.

We calculate LLR and  $NPMI^3$  for each bi-gram, given a word we align the result from LLR and  $NPMI^3$  forming a feature vector of each given word, we truncate it to the top 1500 words.

We build a few matrices of word models, one for each case (verb-verb, noun-noun,...) in each corpus. At the rows we locate the feature vector of the given word, forming a sparse matrix. From this we build a network graph in which a vertex represents a word given by the head of the row on the matrix.

#### B. Uncertainty of “man bites dog” and “dog bites man”

Google Bigrams and en.wikipedia shows that the relation *dog – man* has less uncertainty 1.1715 and 1.1126 respectively, but BNC shows the contrary that relation *man – dog* is less uncertain 1.4038 than *dog – man* 2.8767. The Google Bigrams corpus is a generic approach and is expected to have some error. While en.wikipedia corpus covers a large number of topics and we might argue that the context is too broad. Whereas BNC corpus focuses on the human interaction. Hence we can conclude that BNC and uwJaccard clearly depict the differences among *man-dog*.

To solve the similarity among “*man bites dog*” and “*dog bites man*”, we find uncertainty by summing the uwJaccard from left to right following work by Grefenstette

Table VII: uncertainty among *man* and *dog*, using 3 different corpus

en.wikipedia
$H(\text{dog}) = 1.112468126$ ,
$H(\text{man}) = 2.902715177$
$H(\text{dog}, \text{man}) = 0.000188320$
$H(\text{man}, \text{dog}) = 0.544818488$
$\text{uwJaccard}(\text{dog}, \text{man}) = 1.112656446$
$\text{uwJaccard}(\text{man}, \text{dog}) = 3.447533665$
BNC
$H(\text{dog}) = 2.870717650$
$H(\text{man}) = 1.393004613$
$H(\text{dog}, \text{man}) = 0.006037466$
$H(\text{man}, \text{dog}) = 0.010817212$
$\text{uwJaccard}(\text{dog}, \text{man}) = 2.876755116$
$\text{uwJaccard}(\text{man}, \text{dog}) = 1.403821825$
googleBigram
$H(\text{dog}) = 1.178393207$
$H(\text{man}) = 1.972395788$
$H(\text{dog}, \text{man}) = 0.000640365$
$H(\text{man}, \text{dog}) = 0.000857728$
$\text{uwJaccard}(\text{dog}, \text{man}) = 1.171582092$
$\text{uwJaccard}(\text{man}, \text{dog}) = 1.933390307$

Table VIII: uncertainty among “*man bites dog*” and “*dog bites man*” using Google bigram

googleBigram
$H(\text{man}) = 1.972395788$
$H(\text{dog}) = 1.178393207$
$H(\text{bite}) = 2.608230542$
$\text{uwJaccard}(\text{man}, \text{bite}) = 1.972449690$
$\text{uwJaccard}(\text{bite}, \text{dog}) = 3.011640027$
$\text{uwJaccard}(\text{dog}, \text{bite}) = 1.179965631$
$\text{uwJaccard}(\text{bite}, \text{man}) = 2.692521391$
$\text{man-bites-dog} = 4.984089717$
$\text{dog-bite-man} = 3.872487022$

and Sadrzadeh [6]. We found that “*man bites dog*” is more uncertain than “*dog bites man*” as shown in table 8.

#### IV. CONCLUSION

It is difficult to compare an asymmetric similarity against a symmetric similarity, there is no ground of comparison, one alternative is to compare uJaccard and uwJaccard against Tversky [8] similarity but we will need a stimulus bias, in the case of Tversky a human judgement. More over Tversky similarity does not cover the concept of paths between two objects. On the other hand uwJaccard do consider paths among vertices in a network graph.

We present a novel coefficient similarity, Unilateral Weighted Jaccard Coefficient Similarity (uwJaccard), which is an asymmetric similarity, which measure the similarities base on the the entropy among two words in a network graph. The similarity looks at the uncertainty between word  $W_1$  and word  $W_2$ . It use the cross-entropy to find the uncertainty; been the uncertainty its coefficient.  $\text{uwJaccard}(word_2, word_1) \neq \text{uwJaccard}(word_1, word_2)$ .

We also present that there is less uncertainty in the sentence

“*dog bites man*” than “*man bites dog*” which is proved by uwJaccard under BNC corpus.

#### ACKNOWLEDGMENT

This research was funded in part by ”Fondo para la Innovación, Ciencia y Tecnología - Perú” (FINCyT). We thank the Universidad Católica San Pablo for supporting this research.

#### REFERENCES

- [1] D. McClosky, E. Charniak, and M. Johnson, “Reranking and self-training for parser adaptation,” in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 337–344.
- [2] R. N. Shepard, “Representation of structure in similarity data: Problems and prospects,” *Psychometrika*, vol. 39, no. 4, pp. 373–421, 1974.
- [3] F. Lorrain and H. C. White, “Structural equivalence of individuals in social networks,” *The Journal of Mathematical Sociology*, vol. 1, no. 1, pp. 49–80, 1971.
- [4] M.-M. Deza and E. Deza, *Dictionary of distances*. Elsevier, 2006.
- [5] D. G. Bridge, “De ning and combining symmetric and asymmetric similarity measures,” in *Advances in Case-Based Reasoning (Procs. of 4th European Workshop on Case-Based Reasoning), LNAI-1488*. Citeseer, pp. 52–63.
- [6] E. Grefenstette and M. Sadrzadeh, “Concrete models and empirical evaluations for the categorical compositional distributional model of meaning,” *Computational Linguistics*, vol. 41, no. 1, pp. 71–118, Feb 2015.
- [7] E. W. Holman, “Monotonic models for asymmetric proximities,” *Journal of Mathematical Psychology*, vol. 20, no. 1, pp. 1–15, 1979.
- [8] A. Tversky, “Features of similarity,” *Psychological Review*, vol. 84, pp. 327–352, 1977.
- [9] S. Jimenez, C. Becerra, and A. Gelbukh, “Soft cardinality: A parameterized similarity function for text comparison,” in *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2012, pp. 449–453.
- [10] J. Weeds and D. Weir, “Co-occurrence retrieval: A flexible framework for lexical distributional similarity,” *Computational Linguistics*, vol. 31, no. 4, pp. 439–475, 2005.
- [11] P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity search: the metric space approach*. Springer Science & Business Media, 2006, vol. 32.

- [12] I. Dagan, L. Lee, and F. C. Pereira, “Similarity-based models of word cooccurrence probabilities,” *Machine Learning*, vol. 34, no. 1-3, pp. 43–69, 1999.
- [13] W. Frawley, *International Encyclopedia of Linguistics*. Oxford University Press, 2003, no. v. 4. [Online]. Available: [https://books.google.com.pe/books?id=sl\\_dDVctycgC](https://books.google.com.pe/books?id=sl_dDVctycgC)
- [14] J. Santisteban and J. Tejada Carcamo, “Unilateral jaccard similarity coefficient,” vol. 1393. CEUR Workshop Proceedings, 2015, pp. 23–27.
- [15] J. Santisteban and J. Tejada Carcamo, “Paradigmatic clustering for nlp.” IEEE International Conference on Data Mining Workshops, ICDMW, 2015.
- [16] A. Franz and T. Brants, “All our n-gram are belong to you,” *Google Machine Translation Team*, vol. 20, 2006.
- [17] B. Consortium *et al.*, “The british national corpus, version 3 (bnc xml edition). distributed by oxford university computing services pp the bnc consortium,” 2007.
- [18] J. S. Justeson and S. M. Katz, “Technical terminology: some linguistic properties and an algorithm for identification in text,” *Natural Language Engineering*, vol. 1, pp. 9–27, 3 1995.
- [19] P. A. Pantel, “Clustering by committee,” Ph.D. dissertation, Citeseer, 2003.
- [20] G. Bouma, “Normalized (pointwise) mutual information in collocation extraction,” *Proceedings of GSCL*, pp. 31–40, 2009.

# **Apendice C**

## **Artículo III**

---

*Santisteban, J., Tejada Cárcamo, J. (November-2015). "Paradigmatic Clustering for NLP". IEEE International Conference on Data Mining Workshops, ICDMW.*

## Paradigmatic Clustering for NLP

Julio Santisteban

*Universidad Católica San Pablo*

*Campus Campiña Paisajista s/n*

*Quinta Vivanco, Barrio de San Lázaro*

*Arequipa, Peru*

*Email: jsantisteban@ucsp.edu.pe*

Javier Tejada-Cárcamo

*Universidad Católica San Pablo*

*Campus Campiña Paisajista s/n*

*Quinta Vivanco, Barrio de San Lázaro*

*Arequipa, Peru*

*jtejadac@ucsp.edu.pe*

**Abstract**—How can we retrieve meaningful information from a large and sparse graph?. Traditional approaches focus on generic clustering techniques and discovering dense cumulus in a network graph, however, they tend to omit interesting patterns such as the paradigmatic relations. In this paper, we propose a novel graph clustering technique modelling the relations of a node using the paradigmatic analysis. We exploit node's relations to extract its existing sets of signifiers. The newly found clusters represent a different view of a graph, which provides interesting insights into the structure of a sparse network graph. Our proposed algorithm PaC (Paradigmatic Clustering) for clustering graphs uses paradigmatic analysis supported by a asymmetric similarity, in contrast to traditional graph clustering methods, our algorithm yields worthy results in tasks of word-sense disambiguation. In addition we propose a novel paradigmatic similarity measure. Extensive experiments and empirical analysis are used to evaluate our algorithm on synthetic and real data.

**Keywords**-clustering; paradigmatic; asymmetric similarity;

### I. INTRODUCTION

Data from different domains, such as social networks, bio-informatics, network graph and natural processing language can be modelled as graphs. Each graph contains communities or clusters which represent real groups, each cluster is characterized by a density of inner links than outer links; the main problem is defining what a cluster is and how to identify it. Fortunato and Castellano [1] have identified three main categories to define communities: Local definition, Global definitions and Vertex similarity. A local definition is when communities are grouped together based on their vertices and their immediate neighbours, disregarding the rest of the graph. A global definition is when communities are grouped together based on the analysis of a community with respect to the graph as a whole. Lastly, vertex similarity is defined as communities that are grouped together based on their vertices' similarities, where a quantitative criterion is chosen to evaluate the similarities between each pair of vertices. The main problem in this case is identifying adequate measures of similarity.

Yang and Leskovec [2] have identified 13 main measurements for clustering which are based on internal and external connectivity, a combination of internal and external con-

nnectivity and network model. All algorithms analyse inner and outer links in several ways and measure the degree of correctness of all different communities that are found.

This paper is organized as follow; in the next section we will discuss the Paradigmatic approach for graph clustering. The PaC algorithm is presented in section 3. In section 4 a set of tests are presented, from toy tests to real world tests and finally in section 4 we state the conclusions.

### II. GRAPH CLUSTERING USING PARADIGMATIC STRUCTURE

We propose another sub-structural property between any two vertices in a graph, a paradigmatic property. In this new property vertices are equivalent if they are connected the same way or similarly, this being vertex similarity according to Fortunato and Castellano's definition [1]. We extended concept with an asymmetric similarity coefficient, uJaccard, section II.B.

Paradigmatic analysis seeks to identify the various paradigms (or existing sets of signifiers) which manifest the content of the data. This aspect of structural analysis involves the discovery of the existence of 'underlying' similarity paradigms. Paradigmatic analysis involves comparing and contrasting each of the signifiers that are present in the data. The commutation test can be used in order to identify distinctive signifiers and to define their significance determining whether a change on the level of the signifier leads to a change on the level of the signified [3]. The paradigmatic approach is common in linguistics, social networks and others [4].

There are two kinds of equivalences, structurally equivalent where two vertices in a graph are structurally equivalent if they share many of the same network neighbours. Regular equivalence is more subtle, two regularly equivalent vertices do not necessarily share the same neighbours, but they do have neighbours who are themselves similar [5],[6].

Some vertices are paradigmatically exchangeable if they rely on the same set of edges and their sub-structures are equivalent to a certain degree. For example in some contexts you could use the word *family* to refer to your *friends* and vice versa, the words are not related syntactically or

semantically but in some context are common to interchange the words to extend the meaning. But the similarity from word family to word friend is different to the similarity from word friend to word family, this is because the usage that people gives to the words. We address the similarity in section II.B. Words family and friend do not need to be linked directly by an edge, they might be linked among other vertex or vertices, be able to identify which two words are used interchangeable is the focus of the paradigmatic approach.

PaC focuses on paradigmatic vertices' similarities. Vertices are similar if they are equally shared or have a similar number of edges connected to the same set of vertices directly or indirectly.

In this paper we present a novel graph clustering algorithm based on a paradigmatic approach supported by a bipartite graph [7] and enhanced by modularity. PaC (Paradigmatic Clustering) algorithm is fast and it isn't complex, having a time complexity of  $O(n * \text{avg}(degree)^2)$ , it is also suitable for large graphs, highly parallelized and incremental. PaC is enhanced with a graph's modularity which is done from top to bottom, after an initial cluster has been constructed.

#### A. Paradigmatic Structures

Paradigmatic analysis is a process that identifies entities which are not related directly but are related by their properties, the relatedness among other entities and whether they are interchangeable [3]. In language the reason why we tend to use morphologically unrelated forms in comparative oppositions is to emphasize the semantics which is done by substitution and transposition of words with a similar signifier. Similarity is not defined by a syntactic set of rules but rather by the use of the language. In some cases its use is not grammatically or syntactically correct but it is commonly used.

We defined the signifier as the similarity among vertices of the same cluster, in which not all members of a cluster have the same similarity of relatedness. This is due to the fact that a member of a cluster might belong to more than one cluster with different similarities.

#### B. Paradigmatic Similarity

PaC will use a structural equivalence to cluster vertices. Particularly the Unilateral Jaccard similarity coefficient will be used to find those vertices that have a structural equivalence [8], Unilateral Jaccard is define by:

$$uJaccard(a, c) = \frac{|a \cap c|}{|edges_a|} \quad (1)$$

$$uJaccard(c, a) = \frac{|c \cap a|}{|edges_c|} \quad (2)$$

Where  $a$  and  $c$  are vertices in a network graph.  $a \cap c$  is the intersection of edges between vertices  $a$  and  $c$ , and  $edges_a$  is the number of edges in vertex  $a$ .

#### C. Algorithm

In this section, we will present a novel algorithm to cluster graphs, and a Paradigmatic Clustering algorithm. We propose to use a bipartite graph, to achieve a fast and easy paradigmatic analysis. Our focus is to only compare rows that have at least one column in common, which reduces the space of comparison. Our approach will anchor a row to its closest and most similar row. If we want to extend the clustering level we could anchor a row to more than one row based on a degree of similarity. PaC algorithm can process rows or columns seamlessly, resulting in a bi-clustering algorithm. PaC yielded better results in a sparse matrix due to the low level of connectivity, also its non-dependency allow to run it on a parallelized mode.

The input graph has to be bipartite graph, if the graph is not a bipartite; we transform it on the flight, this is simple task as we start working with a matrix in which column and row will form the two sets of a bipartite graph, knowing that a word might be in both sets we will label as set col-word-xy and row-word-xy. PaC can be applied to the set columns or set rows, getting a binary clustering.

#### D. PaC algorithm

Listing 1. PaC algorithm

```

1 // The graph needs to be bipartite
2 // The graph needs to be an undirected graph
3 procedure PaC (graph, threshold)
4   for each vertex in the graph
5     similaritiesPairs[ ] = uJaccard (Vi,Vn,)
6     // uJaccard two levels out, all possible
7     // paths among Vi and Vn
8     sort similaritiesPairs[ ]
9     group vertices with similar structural
10    equivalence & uJaccard score.
11    grouping up to a threshold
12    clusters [ ] = groups found
13  return clusters[ ]
14  // the grouping is done by selecting the
    elements
15  // up to a particular threshold

```

PaC will use modularity to optimize the found clusters. The use of modularity is not intensive since it is a top to bottom approach. PaC will recalculate the threshold used to identify the cluster cut. In each iteration the modularity can be improved providing a better clustering. The initial cluster found might be or not of interest for the application, but for general purpose and comparison we try to optimize the clustering. The cluster will depend on how similar the clusters need to be, modularity will help to find the optimal cut.

Listing 2. Clustering algorithm

```

1 // we start with modularity of 0
2 // we start with a threshold = 0.5
3 // new modularity is nQ = 0
4 // last modularity is lQ = -1
5 // factor will be the step of reduction = 0.01
6 procedure clus (graph, clusters, threshold)

```

```

7   while nQ > 1Q
8     1Q = nQ
9     threshold[ ] = threshold - factor
10    cluster[ ] = PaC(graph, threshold)
11    nQ = modularity (graph, clusters)
12  return cluster[ ]
13 // threshold is a map <cluster id, threshold >
14 // some clusters will be improved
15 // based on its assigned threshold

```

The uJaccard function has been presented at II.B and equation 1 and 2; and graph modularity introduced by Newman-Girvan [9] is given by:

$$Q = \sum_{i=1}^k (eii - a_i^2) \quad (3)$$

Where  $eii$  = % edges in module  $i$  (probability edge is in module  $i$ ),  $a_i$  = % edges with at least 1 end in module  $i$  (probability a random edge would fall into module  $i$ ).

The first run produces a set of clusters, but are those clusters the optimal cut?. To measure if the produced clusters are the optimal cut we use the graph modularity presented by Newman and Girvan [9] and equation 3. Modularity will be run after PaC, to find out if it has been improved. If not, we stop, otherwise we run PaC again with a lower threshold. Since it is a top-down approach, the calculation of modularity is not intensive like other algorithms; but rather it is used few times to optimize PaC.

The time complexity of our algorithm PaC is  $O(n * avg(degree)^2)$ . We can point out that PaC algorithm is highly parallelized because its similarity calculation does not depend on a previous calculation. Lastly it is incremental since we can add additional vertices and calculate only the vertices involved and not the whole graph.

### III. EXPERIMENTAL EVALUATION

We evaluated the algorithm on toy graphs, synthetic data sets, NLP (Natural Language processing) dataset and real data sets. To prove its efficiency the PaC is compared with the normalized mutual information bench mark in synthetic graphs [10].

#### A. Toy Testing

The use of PaC in toy graphs, as show in Table I, it is clear that the desired cluster from those graphs are obtained by PaC, moreover the state of the art algorithms also find the same clusters.

The toy graphs in Table I are just simple examples to test PaC in particular cases, hence PaC is able to identify cluster properly, in particularly in some special cases like the third toy graph in which cluster 4,5,6 has been identify correctly, also last toy graph in which 2 cluster are identify as expected.

#### B. Real Data Sets

**Zachary's karate club:** The network represents the pattern of friendships among members of a karate club at a

Table I  
TOY GRAPHS CLUSTERED USING PAC

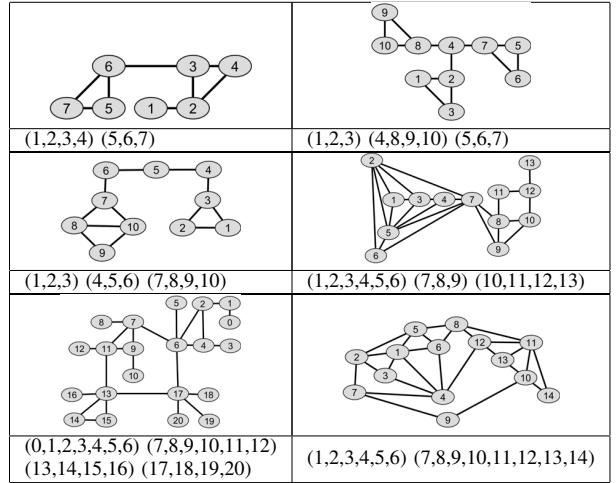


Table II  
RESULTS ON REAL WORLD DATASETS PRODUCED BY PAC. FROM LEFT TO RIGHT, THE NUMBER OF VERTICES N, EDGES E, NORMALIZED MUTUAL INFORMATION NMI, ACTUAL NUMBER OF COMMUNITIES C (BASE LINE) AND PAC NUMBER OF COMMUNITIES.

	N	E	C	NMI	PaC
<b>Zachary's karate club</b>	34	78	2	0.652	4
<b>Dolphins</b>	62	159	2	0.527	5
<b>Football</b>	115	613	12	0.907	15
<b>Les Miserables</b>	77	254	11	0.878	12

North American University [11].

**Football:** A network of American football games [12].

**Dolphins:** A dolphins social network, New Zealand [13].

**Les Miserables:** The network of interactions between major characters in Victor Hugo's sprawling novel of crime and redemption [14].

To measure our findings with a ground-truth graph, we have used the Normalized Mutual Information - NMI [15]. A measure of similarity of partitions borrowed from information theory, which has proven to be reliable [10]. Used by Fortunato Santo [16] to compare two graphs and get a score of the differences between two graphs. NMI of 1 shows equal graphs, 0 shows completely different graphs.

Our findings are shown in Table II, column PaC compared against the baseline column C and the similarity measure in column NMI, we have demonstrating that PaC provides good results.

PaC is able to identify almost correctly clusters in the Football graph with a NMI of 0.9, similarly with Les Miserable graph with a NMI of 0.878, but no so well with Dolphins graph where PaC identifies 5 cluster out of 2; on the Zachary graph PaC has identify 4 out 5, but this is a particular case since the original clustering is 2 but further paper refer up to 4 cluster with some particular vertices that

might belong to different cluster.

### C. Synthetic Data Sets

We have tested PaC, based on a benchmark graph with a built-in community structure. Standard benchmarks, such as the Girvan and Newman, do not account for important features of real network graph, like the fat-tailed distributions of vertex degree and community size [17]. Therefore, we have used a different class of benchmark network graph, in which the distributions of vertex degree and community size are both power laws, with tunable exponents.

To compare the communities found by PaC against a ground-truth network, we use the Normalized Mutual Information [16]. For our testing we followed the settings used by Lancichinetti and Fortunato [18].

To validate the performance of PaC, we compare it against 6 clustering techniques present in iGraph [19], Walktrap Community (wc), Spinglass Community (sc), Leading Eigenvector Community (lec), Fastgreedy Community (fc), Label Propagation Community (lpc) and Multilevel Community (mc).

The mixing parameter in figure 1,2 and 3 is the degree of a cluster is mix with other clusters, 0 means cluster completely separated, where 0.5 means that 50% of the edges in a cluster are linked to other clusters. The realization parameter in figure 1,2 and 3 refers to the number of tests that has been run at each point. such as 1000, 5000 and 10000 tests per point.

PaC out performs mc, fc, lec, sc but it is not the same case for wc and lpc. The wc has a quadratic running time  $O(n^2 \log * n)$  [20] which is not appropriate for very large graphs in comparison to our PaC running time which is  $O(n * \text{avg}(\text{degree})^2)$ .

The lpc has a similar running time as PaC. The differences in performance are due to the fact that each algorithm uses different similarity measures while lpc use a symmetric measure and PaC use an asymmetric coefficient, hence we cannot compare them only based on the outcomes, we also have to consider the type of similarity they use. PaC uses a paradigmatic approach focusing on NLP.

### D. NLP Testing

One of the challenging tasks in NLP is word sense disambiguation, making it even more complex if we use a non-supervised technique. We use PaC to solve this problem and we use empirical analysis to assess our results. PaC will need a graph of words. To build a graph of words we source it from three corpuses, Google n-gram corpus [21], The British National Corpus, version 3, BNC XML Edition [22] and en.wikipedia database backup dumps, each corpus will form an independent set of testing. as shown in table III.

To construct a graph of words we start by constructing a bi-gram model, it was straight forward for Google n-gram

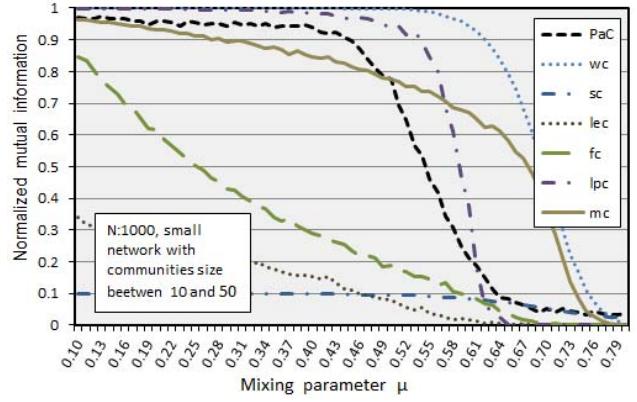


Figure 1. Comparison of PaC with wc, sc, lec, fc, lpc, mc on synthetic data set. The number of vertices is  $N = 5000$ , each point corresponds to 100 graph realizations. PaC clearly outperform sc, lec, fc.

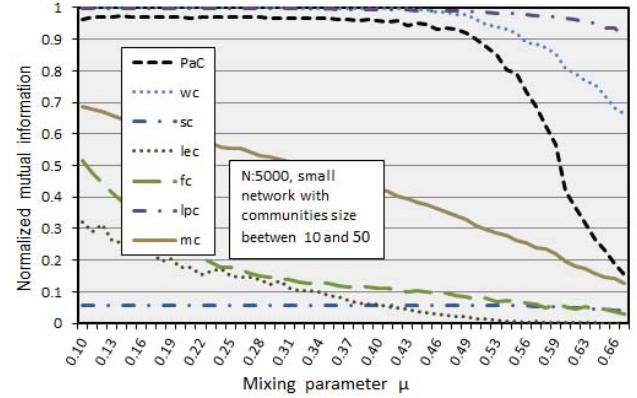


Figure 2. Comparison of PaC with wc, sc, lec, fc, lpc, mc on synthetic data set. The number of vertices is  $N = 1000$ , each point corresponds to 100 graph realizations. PaC clearly outperform sc, lec, fc, mc.

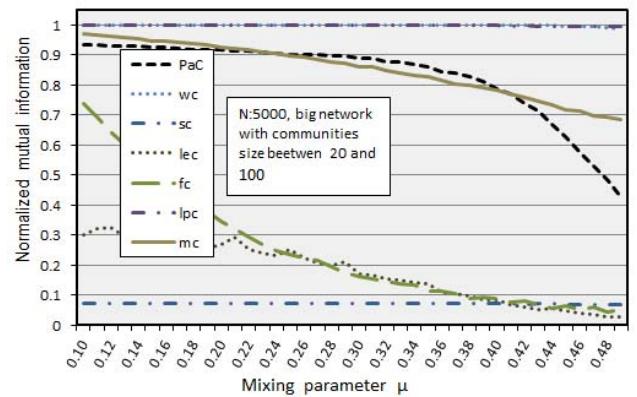


Figure 3. Comparison of PaC with wc, sc, lec, fc, lpc, mc on synthetic data set. The number of vertices is  $N = 5000$ , each point corresponds to 100 graph realizations. PaC clearly outperforms sc, lec, fc, mc.

Table III  
SIZE OF THE USED CORPUS

Corpus	Google bi-gram	BNC	Wikipedia
Size	1.5GB	539MB	11GB
bi-grams	314,843,401	842,162,318	
Words	629,298,853	98,537,224	1,459,026,075
Vocabulary	5,827,772	354,374	3,990,248

as it was in fact a bi-gram dataset. We cleaned up the text on the en.wikipedia and the BNC corpus, and then we run a sentence segmentation script against the text in order to extract sentences. Once the sentences were identified we parse it to discover POS (part of speech), we used the BLLIP parser at T50 [23]. Once we have the parsed sentences we remove stop words (first 1000 more common words) and we apply a lemmatization process to each of the remaining words by doing a look up on WordNet [24]; with a parsed and lemmatized sentence and its POS we start building bi-grams as follows, verb-verb, noun-noun, verb-noun, noun-verb, adverb-verb, verb-adverb, adjective-noun, noun-adjective. We use a window of size n words to the left and/or right of a the working word to build the bi-grams; after a few experiments we use n = 3. This being an extension work of Justeson and Katz [25] and Patrick Pantel [26].

We select two collocation measures from the state of the art, Log-likelihood Ratios LLR [27]. LLR has been selected because it is a good approach to sparse data and it is able to identify rare bi-grams with a low frequency.

The second collocation measure we selected is the Pointwise Mutual Information (PMI) between two words. In information theory, mutual information refers to the mutual information between two random variables rather than between two related events. It is a bad measure of dependence because it does not dependent on the frequency. PMI is bad with sparse data, some words that only occur once, but appear together, score very high PMI. PMI is also bad with word dependence, when two words are perfectly dependent of each other, whenever one occurs; the other occurs, so the rarer the word is, the higher the PMI it gets. Bouma [28] shows several normalized variants of PMI in order to make PMI less sensitive to a low frequency. We have selected the Normalized Pointwise Mutual Information to the power of cube ( $NPMI^3$ ) as it reduces the score for rare words.

We calculate  $LLR$  and  $NPMI^3$  for each bi-gram, given a word we align the result from  $LLR$  and  $NPMI^3$  forming a feature vector of the given word. We use the top 1/3 of the characteristic vector as its collocation ratio is very low and is meaningless on the other 2/3.

We build a few graphs of words, one for each case (verb-verb, noun-noun, verb-noun, noun-verb, adverb-verb, verb-adverb, adjective-noun, noun-adjective) for BNC, en.wikipedia and Google n-gram.

For word sense disambiguation we do not cluster the whole

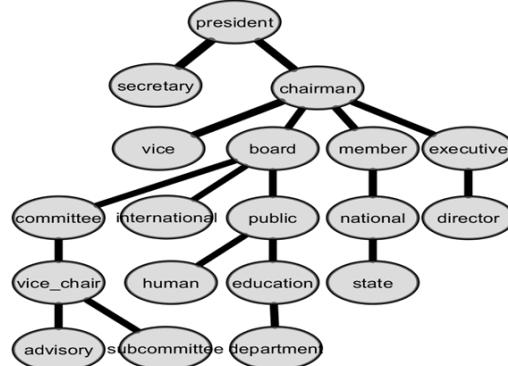


Figure 4. word “chair” extracted from en.wikipedia (noun-noun), it shows the largest cluster.

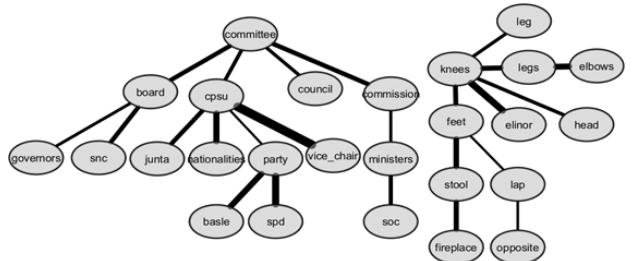


Figure 5. word “chair” extracted from BNC (noun-noun), it shows the largest clusters.

graph, instead we get a subset, the subset which is related to the given word. The sub graph is constructed as follows: we get the feature vector of the given word and load it to the graph, then for each word in the feature vector we load its feature vector to the graph. In a similar way to Fern and Brodley approach [7]. Then we cluster the sub graph using PaC to obtain clusters of senses of a given word.

PaC produces clusters that represent word’s senses, as shown in figures 4 to 8 and Tables IV and V, the uJaccard similarity shows how well related words are. Edge thickness denote the collocation measure given by uJaccard. We notice that each corpus shows different cluster as their contexts are different, as in the case of chair. The results are worthy to consider. Comparing with a base line such as WordNet or another repository will be cumbersome and not very productive because each corpus is completely different and from a different context [29].

PaC produces clusters that represent word-senses, as shown in Table IV and V, the uJaccard similarity shows how well related the words are. Google bi-grams corpus does not have any syntactic information or POS, we just use its frequency to get words collocation and than apply PaC. It is interesting to observe that even with lack of syntax information we get very meaningful word-senses.

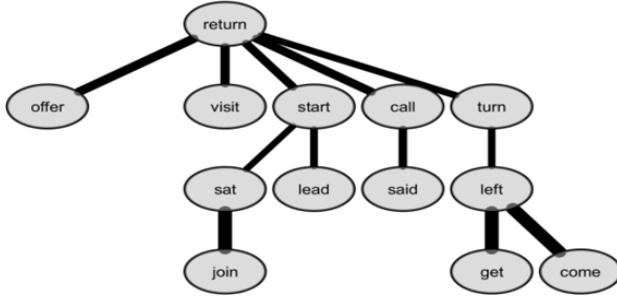


Figure 6. word “chair” extracted from en.wikipedia (noun-verb) it shows largest cluster.

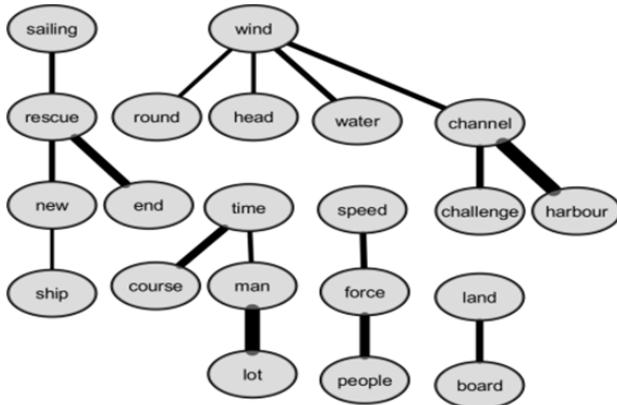


Figure 7. word “sail” extracted from BNC (noun-verb) shows largest clusters.

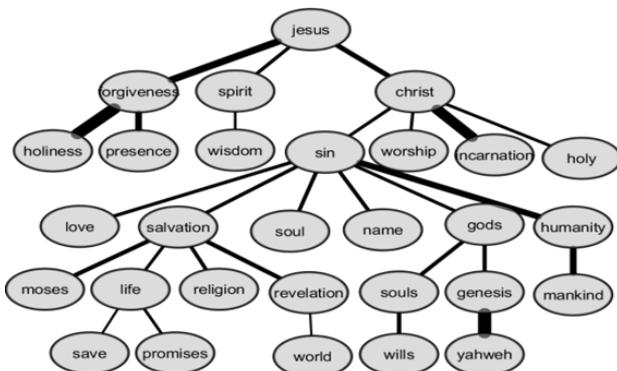


Figure 8. word “god” extracted from BNC( noun-noun) shows largest cluster.

Table IV

SENSES/CLUSTER FOR WORD APPLE, PRODUCED BY PAC USING EN.WIKIPEDIA. THE CLUSTER RELATED TO THE FRUIT IS PRESENTED AND OTHER CLUSTERS ARE GROUPED BY ITS SENSES SUCH AS MUSIC.

macintosh , os, macos, mac, desktop
notebook , laptop, asus, baterry
wireless, usb, memory, remote
release , announce, calendar, make
store, product, user
imac , emac, intel, macbook
cranberry , pectin, cinnamon
juice, cider, martini
software, final, pro, power, hardware, advisor, video, quicktime
music, cinema, aria, garage band, book, deal, vacation, dvd, lossless, itunes, ipod, insight, ipods, archos
valley, spring, orchard, avery, tree, blossom, newton, loop, airport
support, core, developer, weblog, inspiration, iie, iiigs
powerbook, ibook, powermac, xserve

Table V

SENSES FOR THE WORD GOD, PRODUCED BY PAC USING GOOGLE BI-GRAMS; IT IS CLEAR THAT EACH CLUSTER REPRESENTS DIFFERENT ASPECTS RELATED TO GOD.

punish, forgive, almighty, forbid, instruct, spake, ordain
play, free, man, smile, soldier, live, tv
incarnate, bless, forsaken, sake, fear, manifest
told, knew, father, hear
send, save, word, rulesnet
damned, damn, awful, ente
mode, speed, command, channel
brought, bring, giveth, meant, reveal, bring, make, awfuls, gave
desire, set, choose, expect, require, dwell, change, raise, intend, accord
government, church, grant, design, exist, heal, lead

#### IV. CONCLUSION

We have shown how community detection can be approached using a paradigmatic analysis, in which the main idea is to find how similar two non-adjacent vertices are. Paradigmatic analysis is the process that identifies entities which are not related directly, but are similar because of their properties and their relatedness among other entities, similar to the structural equivalence. We introduce a new graph property, a paradigmatic property defined as the set of vertices that are similar if they have a structural equivalence and are interchangeable by substitution and transposition. We also present a novel clustering algorithm PaC, which finds clusters with similar structural equivalence. The structural equivalence is found by using uJaccard. PaC is highly parallelized, incremental and it’s running time is  $O(n * avg(degree)^2)$ . Results show that PaC is able to find word-senses, as well as clusters on a ground-truth network.

#### ACKNOWLEDGMENT

This research was funded in part by ”Fondo para la Innovación, Ciencia y Tecnología - Peru” (FINCyT). We thank the Universidad Católica San Pablo for supporting this research.

## REFERENCES

- [1] S. Fortunato and C. Castellano, “Community structure in graphs,” in *Computational Complexity*. Springer, 2012, pp. 490–512.
- [2] J. Yang and J. Leskovec, “Defining and evaluating network communities based on ground-truth,” *Knowledge and Information Systems*, vol. 42, no. 1, pp. 181–213, 2015.
- [3] F. de Saussure, “Course in general linguistics,” 1959.
- [4] J. T. Cárcamo, H. Calvo, A. Gelbukh, and J. Villegas, “Impacto de recursos léxicos manuales y automáticos en la wsd.” CLEI, 2010.
- [5] F. Lorrain and H. C. White, “Structural equivalence of individuals in social networks,” *The Journal of mathematical sociology*, vol. 1, no. 1, pp. 49–80, 1971.
- [6] D. R. White and K. P. Reitz, “Graph and semigroup homomorphisms on networks of relations,” *Social Networks*, vol. 5, no. 2, pp. 193–234, 1983.
- [7] X. Z. Fern and C. E. Brodley, “Solving cluster ensemble problems by bipartite graph partitioning,” in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 36.
- [8] J. Santisteban and J. Cárcamo, “Unilateral jaccard similarity coefficient,” *Alonso et al.[2]*, pp. 23–27, 2015.
- [9] M. E. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Physical review E*, vol. 69, no. 2, p. 026113, 2004.
- [10] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, “Comparing community structure identification,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.
- [11] W. W. Zachary, “An information flow model for conflict and fission in small groups,” *Journal of anthropological research*, pp. 452–473, 1977.
- [12] M. Girvan and M. E. Newman, “Community structure in social and biological networks,” *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.
- [13] D. Lusseau, “The emergent properties of a dolphin social network,” *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 270, no. Suppl 2, pp. S186–S188, 2003.
- [14] D. E. Knuth, *The Stanford GraphBase: a platform for combinatorial computing*, vol. 37.
- [15] R. M. Fano and D. Hawkins, “Transmission of information: A statistical theory of communications,” *American Journal of Physics*, vol. 29, no. 11, pp. 793–794, 1961.
- [16] S. Fortunato, “Community detection in graphs,” *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.
- [17] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms,” *Physical review E*, vol. 78, no. 4, p. 046110, 2008.
- [18] A. Lancichinetti and S. Fortunato, “Community detection algorithms: a comparative analysis,” *Physical review E*, vol. 80, no. 5, p. 056117, 2009.
- [19] G. Csardi and T. Nepusz, “The igraph software package for complex network research,” *InterJournal, Complex Systems*, vol. 1695, no. 5, pp. 1–9, 2006.
- [20] P. Pons and M. Latapy, “Computing communities in large networks using random walks,” in *Computer and Information Sciences-ISCIS 2005*. Springer, 2005, pp. 284–293.
- [21] A. Franz and T. Brants, “All our n-gram are belong to you,” *Google Machine Translation Team*, vol. 20, 2006.
- [22] B. Consortium *et al.*, “The british national corpus, version 3 (bnc xml edition). distributed by oxford university computing services pp the bnc consortium,” 2007.
- [23] D. McClosky, E. Charniak, and M. Johnson, “Reranking and self-training for parser adaptation,” in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 337–344.
- [24] C. Fellbaum, “Wordnet and wordnets in k. brown (ed.), encyclopedia of language and linguistics (pp. 665–670),” 2005.
- [25] J. S. Justeson and S. M. Katz, “Technical terminology: some linguistic properties and an algorithm for identification in text,” *Natural language engineering*, vol. 1, no. 01, pp. 9–27, 1995.
- [26] P. A. Pantel, “Clustering by committee,” Ph.D. dissertation, Citeseer, 2003.
- [27] T. Dunning, “Accurate methods for the statistics of surprise and coincidence,” *Computational linguistics*, vol. 19, no. 1, pp. 61–74, 1993.
- [28] G. Bouma, “Normalized (pointwise) mutual information in collocation extraction,” *Proceedings of GSCL*, pp. 31–40, 2009.
- [29] E. Agirre, E. Alfonseca, and O. L. De Lacalle, “Approximating hierarchy-based similarity for wordnet nominal synsets using topic signatures,” in *Proceedings of GWC-04, 2nd global WordNet conference*, 2004, pp. 15–22.

# **Apendice D**

## **Artículo IV**

---

*Santisteban, J., Tejada Cárcamo, J. (2016). "Clustering algorithm based on asymmetric similarity and paradigmatic features". IJICA International Journal of Innovative Computing & Applications. Vol 7 Num 4.*

## Clustering algorithm based on asymmetric similarity and paradigmatic features

Julio Santisteban\* and Javier Tejada-Cárcamo

Universidad Católica San Pablo,  
Campus Campiña Paisajista s/n,  
Quinta Vivanco, Barrio de San Lázaro,  
Arequipa, Peru  
Email: jsantisteban@ucsp.edu.pe  
Email: jtejadac@ucsp.edu.pe  
\*Corresponding author

**Abstract:** Similarity measures are essential to solve many pattern recognition problems such as classification, clustering, and information retrieval. Various similarity measures are categorised in both syntactic and semantic relationships. In this paper, we present a novel similarity, unilateral Jaccard similarity coefficient (uJaccard), which does not only take into consideration the space among two points but also the semantics among them. How can we retrieve meaningful information from a large and sparse graph? Traditional approaches focus on generic clustering techniques for network graph. However, they tend to omit interesting patterns such as the paradigmatic relations. In this paper, we propose a novel graph clustering technique modelling the relations of a node using the paradigmatic analysis. Our proposed algorithm paradigmatic clustering (PaC) for graph clustering uses paradigmatic analysis supported by an asymmetric similarity using uJaccard. Extensive experiments and empirical analysis are used to evaluate our algorithm on synthetic and real data.

**Keywords:** clustering algorithm with asymmetric and paradigmatic similarity; clustering algorithm; paradigmatic similarity; asymmetric similarity; similarity.

**Reference** to this paper should be made as follows: Santisteban, J. and Tejada-Cárcamo, J. (2016) ‘Clustering algorithm based on asymmetric similarity and paradigmatic features’, *Int. J. Innovative Computing and Applications*, Vol. 7, No. 4, pp.243–256.

**Biographical notes:** Julio Santisteban is an Associate Researcher at the School of Computer Science at Universidad Católica San Pablo. He received his PhD in Computer Science at the Universidad Nacional de San Agustín, Peru. He received his MSc in Internetworking at the University of Technology Sydney, Australia. His research interests include issues related to data mining, natural language processing and telematic. He is the author of several research studies published at national and international conference proceedings.

Javier Tejada-Cárcamo received his Master’s degree in Computer Science (with honours) in 2005 from the Center for Computing Research (CIC) of the National Polytechnic Institute (IPN), Mexico and PhD in Computer Science (with honours) in 2009 at the same center. Since 2010, he has been an Associate Professor and researcher at San Pablo Catholic University in Arequipa, Peru.

This paper is a revised and expanded version of a paper entitled ‘Paradigmatic clustering for NLP’ presented at IEEE International Conference on Data Mining Workshop (ICDMW), Atlantic City, New Jersey, USA, 14–17 November 2015.

### 1 Introduction

Clustering and classification is one of the most used techniques in computer science and other fields, the key element to group objects in the same community is the measurement of similarity. Similarity is defined by WordNet (Fellbaum, 2005) as “The quality of being similar” and “Similarity in appearance, external or superficial details”, it does not specify if the similarity is symmetric or asymmetric. Understand by common sense that similarity should be symmetric as in a geometrical

model, however, we should not extend the idea of similarity to be symmetric in all the models, such as natural language processing (NLP). In this paper, we present a novel similarity coefficient, Unilateral Jaccard similarity coefficient (uJaccard) which is an extension of Jaccard.

Also, we present a new paradigmatic property for graphs which is extracted from the language field, vertices are interchangeable if they are similar to some degree, the paradigmatic property provides a way to measure their similarity by means of a regular equivalent. The

paradigmatic property will support uJaccard to be used in a graph network.

Having this new approach to measure the similarity by means of an asymmetric similarity and incorporate the graph network's paradigmatic property, we present a novel clustering technique which focuses on NLP but it can be extended to other fields. Paradigmatic clustering (PaC) is a technique that uses an asymmetric and paradigmatic approach to identify communities in a graph network, PaC outperforms the state-of-the-art clustering techniques, PaC is also capable of run in parallel and is a clustering algorithm that supports data streaming, PaC's running time is  $O(n * \text{avg}(degree)^2)$ .

This paper is organised as follow: in the next section, we will discuss the asymmetry of similarity. In Section 3, we discuss the uJaccard. In Section 4, we present the uJaccard experimental evaluation. The PaC algorithm is presented in Section 5. In Section 6, a set of tests are presented, from toy tests to real world tests and finally, in Section 7, we state our conclusions and future work.

## 2 Similarity

Since Euclid to today several similarity measures have been developed, those measures are used in different fields to measure similarity in appearance, external or superficial details, location and others properties of two or more objects. Particularly in the last century with the advances in computer science many new similarity measures have been proposed. Similarity measures are used to compare different kind of data which is fundamentally important for pattern classification, clustering, and information retrieval problems (Duda et al., 2001). Similarity relations have generally been dominated by geometric models in which objects are represented by points in a Euclidean space (Shepard, 1974). Similarity is defined as "Having the same or nearly the same characteristics" (Fellbaum, 2005), while the metric distance is defined as "The property created by the space between two objects or points".

All metric distance functions must satisfy three basic axioms: minimality and equal self-similarity, symmetry, and triangle inequality.

$$d(i, j) = d(j, j) \leq d(i, j) \quad (1)$$

$$d(i, j) = d(j, i) \quad (2)$$

$$d(i, j) + d(j, k) \geq d(i, k) \quad (3)$$

Here for objects  $i, j$  and  $k$ , where  $d()$  is the distance between objects  $i$  and  $j$ . Bridge (1998) argues that there exists empirical evidence of violations against each of the three axioms. Yet, there also exists geometric models of similarity which take asymmetry into account (Nosofsky, 1991). Nosofsky (1991) points out that a number of well-known models for asymmetric proximity data are closely related to the additive similarity and bias model (Holman, 1979). Tversky (1977) has proposed a different model in order to overcome the metric assumption of

geometric models. One of the strengths of contrast models is its capability to explain asymmetric similarity judgements. Tversky's (1977) asymmetry may often be characterised in terms of stimulus bias and determined by the relative prominence of the stimuli.

$$\text{sim}(a, b) = \frac{|A \cap B|}{|A \cap B| + \alpha|A - B| + \beta|B - A|}, \quad (4)$$

$$\alpha, \beta \geq 0$$

Here,  $A$  and  $B$  represent feature sets for the objects  $a$  and  $b$ , respectively; the term in the numerator is a function of the set of shared features, a measure of similarity, and the last two terms in the denominator measure dissimilarity:  $\alpha$  and  $\beta$  are real-number weights; when  $\alpha \neq \beta$ . Jimenez et al. (2012), Weeds and Weir (2005) and Lee et al. (1999) also propose an asymmetric similarity measure based on Tversky's work.

However, all proposals include a stimulus bias, asymmetric similarity judgements, which Tversky refers to as human judgement. Today, similarity measure is deeply embedded into many of the algorithms used for graph classification, clustering and other tasks. Those techniques are leaving aside the semantic of each vertex and its relation among other vertices and edges.

In a direct graph, the similarity from  $U$  to  $Z$  is not the same as the distance from  $Z$  to  $U$ , this due to the intrinsic features of a direct graph. The similarities are different because the channels are dissimilar. According to Shannon's information theory, we could argue that each vertex is a source of energy with an average entropy which is shared among its channels, and while that information flow among the vertex's channels, we need to consider it in the similarity. A similarity does not fit all tasks or cases.

In natural language processing, where the similarity between two words is not symmetric  $\text{sim}(\text{word}_a, \text{word}_b) \neq \text{sim}(\text{word}_b, \text{word}_a)$ . WordNet (Fellbaum, 2005) presents 28 different types of relations; those relations have direction but are not symmetrical, they are not even synonyms because each synonym word has a particular semantic, meaning and usage, but are similar. Hence, if two words have symmetric distance or similarity, those two words are the same.

Paradigmatic is an intrinsic feature in language, it lets the utterer exchange words with other words, words with similar semantics (Sahlgren, 2006). In this paper, we focus on paradigmatic analysis to support our uJaccard.

## 3 Unilateral Jaccard similarity coefficient

### 3.1 Paradigmatic similarity

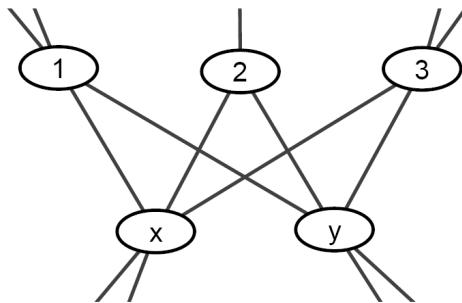
Paradigmatic analysis is a process that identifies entities which are not related directly but are related by their properties, relatedness among other entities and interchangeability (De Saussure et al., 2011). In language, the reason why we tend to use morphologically unrelated forms in comparative oppositions is to emphasise the

semantics, this is done by substitution and transposition of words with a similar signifier. Similarity is not defined by a syntactic set of rules but rather by the use of the language. In some cases, this use is not grammatically or syntactically correct but it is commonly used. We defined the signifier as being the degree of relation among entities of the same group, where not all members of the group have the same degree of relatedness. This is due to the fact that a member of a group might belong to more than one group.

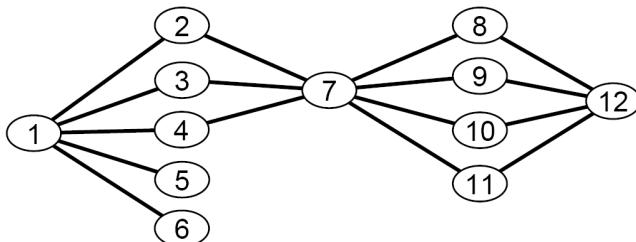
### 3.2 Extended paradigmatic

Two vertices in a graph are structurally equivalent if they share many of the same network neighbours. Figure 1 depicts a structural equivalence between two vertices  $y$  and  $x$  that have the same neighbours. Regular equivalence is more subtle, two regularly equivalent vertices do not necessarily share the same neighbours, but they do have neighbours who are themselves similar (Lorrain and White, 1971; White and Reitz, 1983). We will use regular equivalence as the bases of uJaccard. In Figure 2, vertices 1, 7 and 12 are regular equivalence vertices.

**Figure 1** Structural equivalence



**Figure 2** Toy graph example a



To calculate a paradigmatic similarity, we start with a question, is the similarity coefficient from vertex  $V_a$  to  $V_c$  the same to the similarity coefficient from vertex  $V_c$  to  $V_a$ ? If we argue that both similarity coefficients are the same, we are arguing that the edges from the vertices  $V_a$  and  $V_c$  are the same, and it is clear that that is not usually the case. Thus, both vertices have different sets of edges.

One problem with Tversky (1977) similarity is the estimation for  $\alpha$  and  $\beta$  which are stimulus bias, generally a human factor. Similarly, other similarities which are based on Tversky idea have the same problem. On the other hand, we propose a measure that does not include this bias. We propose a modified version of Jaccard similarity coefficient (5) and (6), uJaccard (7) and (8), used to identify the

similarity coefficient of  $V_a$  to  $V_c$  With respect to vertex  $V_a$ , and to also identify the similarity coefficient of  $V_c$  to  $V_a$  With respect to vertex  $V_c$ .

$$Jaccard(a, c) = \frac{|a \cap c|}{|a \cup c|} \quad (5)$$

$$Jaccard(c, a) = \frac{|c \cap a|}{|c \cup a|} \quad (6)$$

$$uJaccard(a, c) = \frac{|a \cap c|}{\text{degree}(a)} \quad (7)$$

$$uJaccard(c, a) = \frac{|c \cap a|}{\text{degree}(c)} \quad (8)$$

where  $a$  and  $c$  are vertices in a network graph.  $a \cap c$  is the intersection of edges between vertices  $a$  and  $c$ .  $\text{degree}(a)$  is the degree of vertex  $a$  and  $\text{degree}(c)$  is the degree of vertex  $c$ .

If uJaccard is close to 0, it means that they are not similar at all. The objective of using uJaccard is to identify how similar a vertex is to other vertices in relation to itself. uJaccard could be calculated among two connected vertices, uJaccard could also be calculated among vertices that are not connected directly, but which are connected by in-between vertices. The number of in-between vertices could be from 1 to  $n$ , we do not recommend a deep comparison since the semantics of the vertex loosest its meaning. Hence,  $\max(n) = 3$ , it is suggested for NLP. For the calculations, we do not consider the number of in-between vertices since we focus on the information flow and not the information transformation carried out on the intermediate vertices.

## 4 The uJaccard experimental evaluation

### 4.1 Toy testing

Using similarity uJaccard (7) and (8), we can build a paradigmatic approach to group vertices. Figure 2 shows a toy graph with 12 vertices and 16 edges, following the paradigmatic analysis, we can determine that vertices 12 and 7 belong to group  $P$  because they have the same number of edges to a same set of vertices. Vertex 1 also belongs to group  $P$  because vertex 1 has 3 of the 5 edges, the same as vertex 7, the degree of membership of vertex 1 is lower than vertices 7 and 12 because vertex 1 has other edges that are not shared by vertices 7 or 12. In the same manner, we can determine that vertex 8, 9, 10 and 11 belong to group  $Q$  because they have an equal number of edges to the same set of vertices. Similarly, vertices 2, 3 and 4 belong to group  $R$ , and vertices 5 and 6 belong to group  $O$ . In this example, we can easily identify the paradigmatic approach, where two or more vertices belong to the same group if they have the same or similar neighbours, but the neighbours in turn belong to another group.

Following the uJaccard similarity and the paradigmatic approach, the results of the graph in Figure 2 are shown in

Table 1. We notice that uJaccard similarity provides better information of similarity than Jaccard; this is because uJaccard considers the notion of unilateral similarity. Table 1 shows three toy graphs, in which we present a comparison between Jaccard and uJaccard. As shown in uJaccard provides a unilateral similarity improving the symmetric similarity Jaccard.

**Table 1** The uJaccard and Jaccard calculation from Figure 2

uJaccard (V1, V7) = 3/5 = 0.600
uJaccard (V7, V1) = 3/7 = 0.428
uJaccard (V7, V12) = 4/7 = 0.571
uJaccard (V12, V7) = 4/4 = 1.000
Jaccard (V1, V7) = 3/9 = 0.333
Jaccard (V7, V1) = 3/9 = 0.333
Jaccard (V7, V12) = 4/7 = 0.571
Jaccard (V12, V7) = 4/7 = 0.571

**Table 2** Test uJaccard in toy graphs

	uJaccard
	sim(2, 5) = 4/4 sim(5, 2) = 4/6 Jaccard sim(2, 5) = 4/6 sim(5, 2) = 4/6
	uJaccard sim(7, 5) = 1/3 sim(5, 7) = 1/1 Jaccard sim(7, 5) = 1/3 sim(5, 7) = 1/3
	uJaccard sim(2, 4) = 3/4 sim(4, 2) = 3/5 Jaccard sim(2, 4) = 3/6 sim(4, 2) = 3/6

Table 2 shows three toy graphs, in which we present a comparison among Jaccard and uJaccard. As shown in uJaccard provide an improving similarity over Jaccard.

#### 4.2 Cut a graph

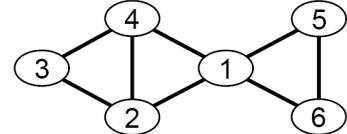
In graph theory, a cut is a partition of the vertices of a graph into two disjoint subsets. There are many techniques and algorithms to cut a graph, but in some cases there are graphs that are difficult to cut, due to their symmetric distribution of vertices.

It is shown in Figure 3 that node 1 might belong to clusters {2, 3, 4} or clusters {5, 6}; to resolve this problem, we use uJaccard similarity measure to find the similarity of

node 1 to other nodes. Table 3 shows that similarities from node 1 to other nodes 1 level deep are the same, so we could not allocate node 1 to a particular cluster.

Table 3 also shows that similarities from node 1 to other node 2 levels deep, in which uJaccard(1, 3) has a strong similarity over the rest. We could conclude that node 1 belong to clusters {2, 3, 4}.

**Figure 3** Toy graph example b

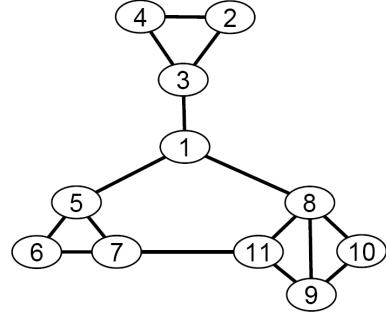


**Table 3** Cutting a graph (Figure 3) using uJaccard

	1 level deep	2 levels deep	
uJaccard(1, 4)	1/4	uJaccard(1, 2)	1/4
uJaccard(1, 2)	1/4	uJaccard(1, 3)	2/4
uJaccard(1, 5)	1/4	uJaccard(1, 4)	1/4
uJaccard(1, 6)	1/4	uJaccard(1, 5)	1/4
-	-	uJaccard(1, 6)	1/4

In Figure 4, also node 1 might belong to clusters {2, 3, 4} or clusters {5, 6, 7} or clusters {8, 9, 10, 11}; this is where uJaccard comes in, being able to solve this problem. Table 4 shows result of similarities from node 1 to all other nodes on the network in different levels deep. Clusters {8, 9, 10, 11} present the highest number of strong similarities, therefore, we can conclude that node 1 belongs to clusters {8, 9, 10, 11}.

**Figure 4** Toy graph example c



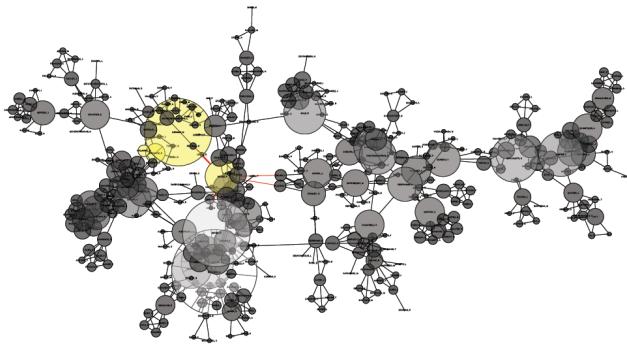
**Table 4** Cutting a graph (Figure 4) using uJaccard

	2 levels deep	3 levels deep	
uJaccard(1, 4)	1/3	uJaccard(1, 4)	1/3
uJaccard(1, 2)	1/3	uJaccard(1, 2)	1/3
uJaccard(1, 6)	1/3	uJaccard(1, 6)	1/3
uJaccard(1, 7)	1/3	uJaccard(1, 7)	2/3
uJaccard(1, 9)	1/3	uJaccard(1, 9)	2/3
uJaccard(1, 11)	1/3	uJaccard(1, 11)	2/3
uJaccard(1, 10)	1/3	uJaccard(1, 10)	1/3

### 4.3 Social network

We tested uJaccard against two social network graphs; the first is the co-authorship network of scientists Newman (2006), the second is the network of Hollywood's actors. The second is the network of Hollywood's actors , we used the public Internet Movie Database's (IMDB). Edge connections between actors were established according to joint participation in at least one movie or TV episode.

**Figure 5** Coauthorship network of scientists, selected nodes belong to scientists Newman, Callaway, Strogatz, Holme (see online version for colours)



**Table 5** The uJaccard calculation from Figure 5, in search paradigmatic scientists to Newman

	2 levels deep	3 levels deep	4 levels deep	
<i>Scientist Newman is similar to:</i>				
Callaway	0.15	Strogatz	1.63	Strogatz
Strogatz	0.15	Holme	1.59	Callaway
Watts	0.15	Kleinberg	1.59	Watts
Hopcroft	0.11	Sole	1.59	Kleinberg
<i>Scientists that are similar to Newman:</i>				
Adler	0.33	Aberg	0.50	Aberg
Aharony	0.33	Adler	0.66	Adler
Aleksiejuk	0.50	Aharony	0.66	Aharony
Ancelmeyers	0.66	Alava	0.50	Alava
Araujo	0.33	Albert	0.10	Albert

The first network is the co-authorship network of scientists working on network theory and experiments, compiled by Newman (2006). We want to find the top scientists that Newman is similar to or that have paradigmatic similarity. As shown in Table 5, the three most of *Newman's* paradigmatic similar scientists are *Callaway*, *Strogatz* and *Holme*. On the other hand, the top 3 scientists that are similar to *Newman* are *Adler*, *Aberg* and *Aharony*. uJaccard has been calculated in 2, 3 and 4 levels deep away from *Newman*. scientists *Newman* is more similar to *Strogatz* but the most similar scientist to new *Newman* is *Adler* and not *Strogatz*, even that *Strogatz* most similarity is toward *Newman*.

For the second network, we created the second social network of Hollywood's actors, we based on The Internet Movie Database. We download actors and actresses data,

which includes title of movies in which they worked; we also download a list of top 1,000 and top 250 actors and actresses. The network is composed of nodes representing actors and actresses, and vertices are the movies in which those actors worked together. A node is created for every person, with their names as the key, when two people are in the same movie; a vertex is created between their nodes. The first network presents 1,000 top actors and actresses who also work in 41,719 movies with a total 113,478 edges. The second network presents 250 actors and actresses who work in 15,831 movies with a total of 14,096 edges. For this test, we remove duplicated edges.

- from a given *actor A*
- we search for actors that *actor A* is similar to
- from the *actor A*'s similar actor list we get the most similar *actor B*
- we search for actors that *actor B* is similar to
- this is done to analyse if *actor A* and *actor B* are reciprocally similar
- then we look for actors that are most similar to *actor A*
- we do this on the network top 250 actors and top 1,000 actors.

**Table 6** The uJaccard similarity among top 1,000 and top 250 actor and actresses, searching paradigmatic similar actor

Top 1,000 actors, cruise tom is similar to:			
Roberts Julia			
Roberts Julia	0.405	Travolta John	0.418
Hanks Tom	0.401	Hanks Tom	0.412
Jackson Samuel	0.399	Jackson Samuel	0.407
Douglas Michael	0.397	Cruise Tom	0.399
Eastwood Clint	0.393	Spacey Kevin	0.399
Top 250 actors, cruise tom is similar to:			
hanks tom			
Hanks Tom	0.430	Douglas Michael	0.425
Douglas Michael	0.420	Cruise Tom	0.421
Eastwood Clint	0.420	Jackson Samuel	0.418
Spacey Kevin	0.413	Travolta John	0.414
Jackson Samuel	0.410	Spacey Kevin	0.411
Who is similar to cruise tom:			
top 1000 actors		top 250 actors	
Ledger Heath	0.490	Allen Joan	0.496
Bacon Kevin	0.488	Balk Fairuza	0.495
Crowe Russell	0.482	Bello Maria	0.488
Gibson Mel	0.482	Collins Pauline	0.487
Benigni Roberto	0.482	Aiello Danny	0.487

The results of the search on the network of top 250 actors and top 1,000 actors, using uJaccard and the paradigmatic approach are presented in Tables 6 and 7. In Table 6, we

focus in *Tom Cruise*, we found that *Tom Cruise* is most similar to *Julia Roberts* but *Julia Roberts* is most similar to *John Travolta*, *Tom Cruise* is third in *Julia Roberts'* similarity list. Clearly, there is not a symmetric similarity among *Julia Roberts* and *Tom Cruise*. Moreover *Julia Roberts* is not the most similar toward *Tom Cruise*, the most similar towards *Tom Cruise* is *Heath Ledger*. Hence, this confirms that uJaccard helps to identify similarities, particularly asymmetric similarities. Table 6 also shows similar scenario among *Tom Cruise*, *Tom Hanks* and *Joan Allen* in the network of top 250 actors and actresses, this confirm the usability of uJaccard.

**Table 7** The uJaccard similarity among top 250 actor and actresses, searching paradigmatic actor, in 2 and 3 levels deep

Top 250 actors, are similar to:			
Quinn Anthony		Nicholson Jack	
Hanks Tom	0.451	Hanks Tom	0.436
Jackson Samuel	0.443	Eastwood Clint	0.429
Lemmon Jack	0.443	Travolta John	0.417
Cruise Tom	0.435	Williams Robin	0.417
De Niro Robert	0.435	Douglas Michael	0.414
Who are similar to Quinn Anthony:			
1 level deep		2 levels deep	
Banderas Antonio	0.366	Banderas Antonio	21.866
Bardem Javier	0.350	Benigni Roberto	20.758
Martin Steve	0.333	Burns George	20.565
Goodman John	0.320	Baldwin Alec	20.413
Allen Woody	0.285	Mcqueen Steve	20.244
Who are similar to Nicholson Jack:			
1 level deep		2 levels deep	
Greene Graham	0.484	Banderas Antonio	41.477
Bronson Charles	0.482	Bacon Kevin	41.133
Brody Adrien	0.480	Baldwin Alec	41.0862
Bale Christian	0.476	Bronson Charles	40.982
Baldwin Alec	0.474	Benigni Roberto	40.948

In Table 7, we use the network of top 250 actors and actresses and we focus on *Anthony Quinn* and *Jack Nicholson*. We start by searching for actors that *Anthony Quinn* is similar to, then we search for actors that are most similar to *Anthony Quinn* in 1 and 2 levels deep. We notice that *Anthony Quinn* is most similar to *Tom Hanks* but most similar actor to *Anthony Quinn* is *Antonio Banderas*, while *Anthony Quinn* is the 153rd most similar for *Tom Hanks*. *Antonio Banderas* is most similar to *Samuel Jackson* and not to *Anthony Quinn*, while *Anthony Quinn* is the 53rd most similar for *Antonio Banderas*. Therefore, we could conclude that *Anthony Quinn* and *Tom Hanks* are not symmetric similar rather they are asymmetric similar. Table 7 also shows similar scenario for *Jack Nicholson*.

## 5 The PaC

### 5.1 Clustering

Data from different domains, such as social networks, bio-informatics, network graph and natural processing language can be modelled as graphs. Each graph contains communities or clusters which represent real groups, each cluster is characterised by a density of inner links than outer links; the main problem is defining what a cluster is and how to identify it. Fortunato and Castellano (2012) have identified three main categories to define communities: local definition, global definitions and vertex similarity. A local definition is when communities are grouped together based on their vertices and their immediate neighbours, disregarding the rest of the graph. A global definition is when communities are grouped together based on the analysis of a community with respect to the graph as a whole. Lastly, vertex similarity is defined as communities that are grouped together based on their vertices' similarities, where a quantitative criterion is chosen to evaluate the similarities between each pair of vertices. The main problem in this case is identifying adequate measures of similarity (Hochreiter and Waldhauser, 2013; Real et al., 2014).

Yang and Leskovec (2015) have identified 13 main measurements for clustering which are based on internal and external connectivity, a combination of internal and external connectivity and network model. All algorithms analyse inner and outer links in several ways and measure the degree of correctness of all different communities that are found.

### 5.2 Paradigmatic property

We propose another sub-structural property between any two vertices in a graph, a paradigmatic property. In this new property, vertices are regular equivalent if they are connected the same way or similarly, this being vertex similarity according to definition provide by Fortunato and Castellano (2012). We extended concept with an asymmetric similarity coefficient, uJaccard.

Paradigmatic analysis seeks to identify the various paradigms (or existing sets of signifiers) which manifest the content of the data. This aspect of structural analysis involves the discovery of the existence of 'underlying' similarity paradigms. Paradigmatic analysis involves comparing and contrasting each of the signifiers that are present in the data. The commutation test can be used in order to identify distinctive signifiers and to define their significance determining whether a change on the level of the signifier leads to a change on the level of the signified (De Saussure et al., 2011). The paradigmatic approach is common in linguistics, social networks and others (Tejada-Cárcamo et al., 2010).

PaC focuses on paradigmatic vertices' similarities. Vertices are similar if they are equally shared or have a similar number of edges connected to the same set of vertices directly or indirectly.

In this paper, we present a novel graph clustering algorithm based on a paradigmatic approach supported by a bipartite graph (Fern and Brodley, 2004) and enhanced by modularity. PaC algorithm is fast and it is not complex, having a time complexity of  $O(n * \text{avg}(\text{degree})^2)$ , it is also suitable for large graphs, highly parallelised, incremental and data stream clustering. PaC is enhanced with a graph's modularity which is done from top to bottom, after an initial cluster has been constructed.

### 5.3 Paradigmatic structures

Paradigmatic analysis is a process that identifies entities which are not related directly but are related by their properties, the relatedness among other entities and whether they are interchangeable (De Saussure et al., 2011). In language, the reason why we tend to use morphologically unrelated forms in comparative oppositions is to emphasise the semantics which is done by substitution and transposition of words with a similar signifier. Similarity is not defined by a syntactic set of rules but rather by the use of the language. In some cases, its use is not grammatically or syntactically correct but it is commonly used.

We based PaC on the principle of regular equivalence, some vertices are paradigmatically exchangeable if they rely on the same set of edges and their sub-structures are regular equivalent to a certain degree. For example, in some contexts, you could use the word *family* to refer to your *friends* and vice versa, the words are not related syntactically or semantically but in some context are common to interchange the words to extend the meaning. But the similarity from word family to word friend is different to the similarity from word friend to word family, this is because the usage that people gives to the words. Words family and friend do not need to be linked directly by an edge, they might be linked among other vertex or vertices, be able to identify which two words are used interchangeable is the focus of the paradigmatic approach.

We defined the signifier as the similarity among vertices of the same cluster, in which not all members of a cluster have the same similarity of relatedness. This is due to the fact that a member of a cluster might belong to more than one cluster with different similarities.

### 5.4 Paradigmatic similarity

PaC will use a structural equivalence to cluster vertices. Particularly, the uJaccard will be used to find those vertices that have a regular equivalence (Santisteban and Tejada-Carcamo, 2015b), unilateral Jaccard (7) (8).

### 5.5 Algorithm

In this section, we will present a novel algorithm to cluster graphs, and a PaC algorithm. We propose to use a bipartite graph, to achieve a fast and easy paradigmatic analysis. Our focus is to only compare rows that have at least one column in common, which reduces the space of comparison. Our approach will anchor a row to its closest and most similar

row. If we want to extend the clustering level we could anchor a row to more than one row based on a degree of similarity. PaC algorithm can process rows or columns seamlessly, resulting in a bi-clustering algorithm. PaC yielded better results in a sparse matrix due to the low level of connectivity, also its non-dependency allow to run it on a parallelised mode (Santisteban and Tejada-Carcamo, 2015a).

The input graph has to be bipartite graph, if the graph is not a bipartite; we transform it on the flight, this is simple task as we start working with a matrix in which column and row will form the two sets of a bipartite graph, knowing that a word might be in both sets we will label as set col-word-xy and row-word-xy. PaC can be applied to the set columns or set rows, getting a binary clustering.

**Listing 1** PaC algorithm (see online version for colours)

---

```
// The graph needs to be bipartite
// The graph needs to be an undirected graph
procedure PaC (graph, threshold)
    for each vertex in the graph
        similaritiesPairs[ ] = uJaccard (Vi, Vn)
        // uJaccard two levels out, all possible
        // paths among Vi and Vn
        sort similaritiesPairs[ ]
        group vertices with similar structural
        equivalence & uJaccard score.
        grouping up to a threshold
        clusters [ ] = groups found
    return clusters[ ]
    // the grouping is done by selecting the elements
    // up to a particular threshold
```

---

**Listing 2** Clustering algorithm (see online version for colours)

---

```
// we start with modularity of 0
// we start with a threshold = 0.5
// new modularity is nQ = 0
// last modularity is lQ = -1
// factor will be the step of reduction = 0.01
procedure clus (graph, clusters, threshold)
    while nQ > lQ
        lQ = nQ
        threshold[ ] = threshold - factor
        cluster[ ] = PaC(graph, threshold)
        nQ = modularity (graph, clusters)
    Return cluster[ ]
    // threshold is a map <cluster id, threshold >
    // some clusters will be improved
    // based on its assigned threshold
```

---

## 5.6 PaC algorithm

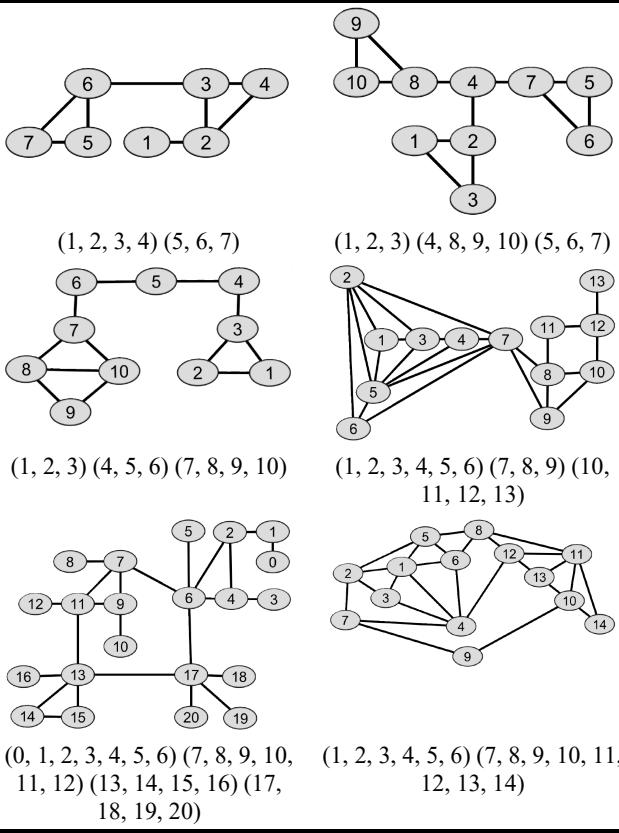
PaC will use modularity to optimise the found clusters. The use of modularity is not intensive since it is a top to bottom approach. PaC will recalculate the threshold used to identify the cluster cut. In each iteration, the modularity can be improved providing a better clustering. The initial cluster found might be or not of interest for the application, but for general purpose and comparison we try to optimise the clustering. The cluster will depend on how similar the clusters need to be, modularity will help to find the optimal cut.

The uJaccard function has been presented in equations (7) and (8); and graph modularity introduced by Newman-Girvan (Newman and Girvan, 2004) is given by:

$$Q = \sum_{i=1}^k (e_{ii} - a_i^2) \quad (9)$$

where  $e_{ii}$  = % edges in module  $i$  (probability edge is in module  $i$ ),  $a_i^2$  = % edges with at least 1 end in module  $i$  (probability a random edge would fall into module  $i$ ).

**Table 8** Toy graphs clustered using PaC



The first run produces a set of clusters, but are those clusters the optimal cut? To measure if the produced clusters are the optimal cut we use the graph modularity presented by Newman and Girvan (2004) and equation (9). Modularity will be run after PaC, to find out if it has been improved. If not, we stop, otherwise we run PaC again with a lower threshold. Since it is a top-down approach, the calculation

of modularity is not intensive like other algorithms; but rather it is used few times to optimise PaC.

The time complexity of our algorithm PaC is  $O(n * avg(degree)^2)$ . We can point out that PaC algorithm is highly parallelised because its similarity calculation does not depend on a previous calculation. Lastly, it is incremental since we can add additional vertices and calculate only the vertices involved and not the whole graph.

## 6 PaC experimental evaluation

We evaluated the algorithm on toy graphs, synthetic datasets, NLP dataset and real datasets. To prove its efficiency, the PaC is compared with the normalised mutual information (NMI) bench mark in synthetic graphs (Danon et al., 2005).

### 6.1 Toy testing

The use of PaC in toy graphs, as shown in Table 8, it is clear that the desired cluster from those graphs are obtained by PaC, moreover the state-of-the-art algorithms also find the same clusters.

The toy graphs in Table 8 are just simple examples to test PaC in particular cases, hence PaC is able to identify cluster properly, in particularly in some special cases like the third toy graph in which clusters 4, 5, 6 has been identify correctly, also last toy graph in which two clusters are identify as expected.

**Table 9** Results on real world datasets produced by PaC

	<i>N</i>	<i>E</i>	<i>C</i>	<i>NMI</i>	<i>PaC</i>
Zachary's karate club	34	78	2	0.652	4
Dolphins	62	159	2	0.527	5
Football	115	613	12	0.907	15
Les Miserables	77	254	11	0.878	12

Note: From left to right, the number of vertices *N*, edges *E*, NMI, actual number of communities *C* (base line) and PaC number of communities.

### 6.2 Real datasets

To be able to validate our proposal, we select some of the most common graphs that represent actual interactions such as the social human interaction, dolphin social network and characters interaction on the literature. This will help us to understand the relevance of our proposal with the reality.

- *Zachary's karate club*: The network represents the pattern of friendships among members of a karate club at a North American University (Zachary, 1977).
- *Football*: A network of American football games (Girvan and Newman, 2002).

- *Dolphins*: A dolphin's social network, New Zealand (Lusseau, 2003).
- *Les Misérables*: The network of interactions between major characters in Victor Hugo's sprawling novel of crime and redemption (Knuth, 1993).

To measure our findings with a ground-truth graph, we have used the NMI (Fano and Hawkins, 1961). A measure of similarity of partitions borrowed from information theory, which has proven to be reliable (Danon et al., 2005). Used by Fortunato (2010) to compare two graphs and get a score of the differences between two graphs. NMI of 1 shows equal graphs, 0 shows completely different graphs.

Our findings are shown in Table 9, column PaC compared against the baseline column C and the similarity measure in column NMI, we have demonstrating that PaC provides good results.

PaC is able to identify almost correctly clusters in the Football graph with a NMI of 0.9, similarly with Les Miserable graph with a NMI of 0.878, but no so well with Dolphins graph where PaC identifies five cluster out of two; on the Zachary graph PaC has identify four out five, but this is a particular case since the original clustering is 2 but further paper refer up to four cluster with some particular vertices that might belong to different cluster.

### 6.3 Synthetic datasets

We have tested PaC, based on a benchmark graph with a built-in community structure. Standard benchmarks, such as the Girvan and Newman (2002), do not account for important features of real network graph, like the fat-tailed distributions of vertex degree and community size (Lancichinetti et al., 2008). Therefore, we have used a different class of benchmark network graph, in which the distributions of vertex degree and community size are both power laws, with tunable exponents.

To compare the communities found by PaC against a ground-truth network, we use the NMI (Fortunato, 2010). For our testing, we followed the settings used by Lancichinetti and Fortunato (2009).

To validate the performance of PaC, we compare it against six clustering techniques present in iGraph (Csardi and Nepusz, 2006), walktrap community (wc), spinglass community (sc), leading eigenvector community (lec), fastgreedy community (fc), label propagation community (lpc) and multilevel community (mc).

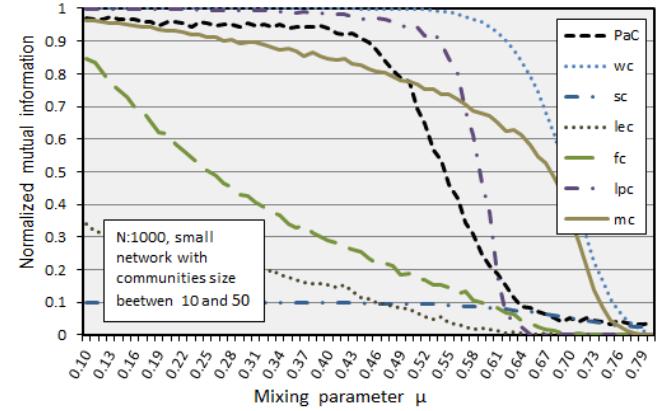
The mixing parameter in Figures 1, 2 and 3 is the degree of a cluster is mix with other clusters, 0 means cluster completed separated, where 0.5 means that 50% of the edges in a cluster are linked to other clusters. The realisation parameter in Figures 1, 2 and 3 refers to the number of tests that has been run at each point, such as 1,000, 5,000 and 10,000 tests per point.

PaC out performs mc, fc, lec, sc but it is not the same case for wc and lpc. The wc has a quadratic running time  $O(n^2 \log * n)$  (Pons and Latapy, 2005) which is not

appropriate for very large graphs in comparison to our PaC running time which is  $O(n * \text{avg}(\text{degree})^2)$ .

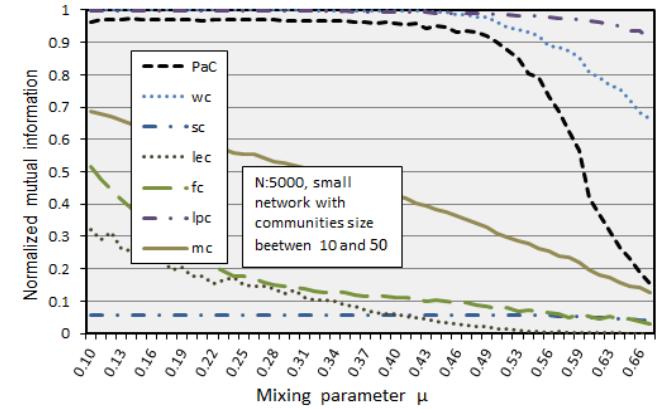
The lpc has a similar running time as PaC. The differences in performance are due to the fact that each algorithm uses different similarity measures while lpc use a symmetric measure and PaC use an asymmetric coefficient, hence, we cannot compare them only based on the outcomes, we also have to consider the type of similarity they use. PaC uses a paradigmatic approach focusing on NLP.

**Figure 6** Comparison of PaC with wc, sc, lec, fc, lpc, mc on synthetic dataset (see online version for colours)



Notes: The number of vertices is  $N = 5,000$ , each point corresponds to 100 graph realisations. PaC clearly outperforms sc, lec, and fc.

**Figure 7** Comparison of PaC with wc, sc, lec, fc, lpc, mc on synthetic dataset (see online version for colours)



Notes: The number of vertices is  $N = 1,000$ , each point corresponds to 100 graph realisations. PaC clearly outperforms sc, lec, fc, and mc.

### 6.4 NLP testing

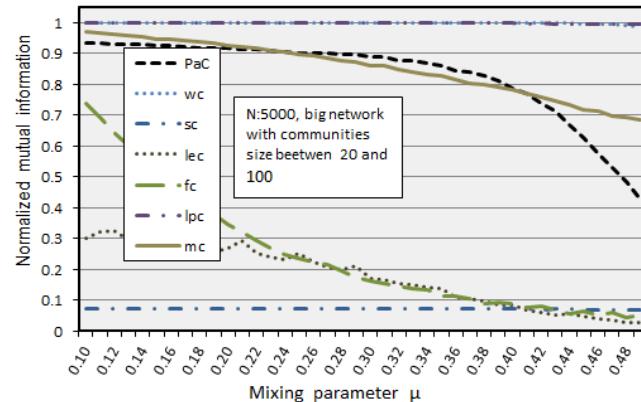
One of the challenging tasks in NLP is word sense disambiguation, making it even more complex if we use a non-supervised technique. We use PaC to solve this problem and we use empirical analysis to assess our results. PaC will need a graph of words.

To build a graph of words, we source it from three corpuses, Google n-gram corpus (Franz and Brants, 2006),

The British National Corpus, version 3, BNC XML Edition (Consortium et al., 2007) and en.wikipedia database backup dumps, each corpus will form an independent set of testing.

To construct a graph of words we start by constructing a bi-gram model, it was straight forward for Google n-gram as it was in fact a bi-gram dataset. We cleaned up the text on the en.wikipedia and the BNC corpus, and then we run a sentence segmentation script against the text in order to extract sentences. Once the sentences were identified we parse it to discover part of speech (POS), we used the BLLIP parser at T50 (McClosky et al., 2006). Once we have the parsed sentences we remove stop words (first 1,000 more common words) and we apply a lemmatisation process to each of the remaining words by doing a look up on WordNet (Fellbaum, 2005); with a parsed and lemmatised sentence and its POS we start building bi-grams as follows, verbverb, noun-noun, verb-noun, noun-verb, adverb-verb, verb-adverb, adjective-noun, noun-adjective. We use a window of size  $n$  words to the left and/or right of a working word to build the bi-grams; after a few experiments we use  $n = 3$ . This is being an extension work of Justeson and Katz (1995) and Pantel (2003).

**Figure 8** Comparison of PaC with wc, sc, lec, fc, lpc, mc on synthetic dataset (see online version for colours)



Notes: The number of vertices is  $N = 5,000$ , each point corresponds to 100 graph realisations. PaC clearly outperforms sc, lec, fc, and mc.

**Table 10** Size of the used corpus

Corpus	Google bi-gram	BNC	Wikipedia
Size	1.5 GB	539 MB	11 GB
bi-grams	314,843,401	842,162,318	
Words	629,298,853	98,537,224	1,459,026,075
Vocabulary	5,827,772	354,374	3,990,248

We select two collocation measures from the state-of-the-art, log-likelihood ratios (LLR) (Dunning, 1993). LLR has been selected because it is a good approach to sparse data and it is able to identify rare bi-grams with a low frequency.

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)} \quad (10)$$

$$\log \lambda = \log \frac{b(c_{12}, c_1, p)b(c_2 - c_{12}, N - c_1, p)}{b(c_{12}, c_1, p_1)b(c_2 - c_{12}, N - c_1, p_2)} \quad (11)$$

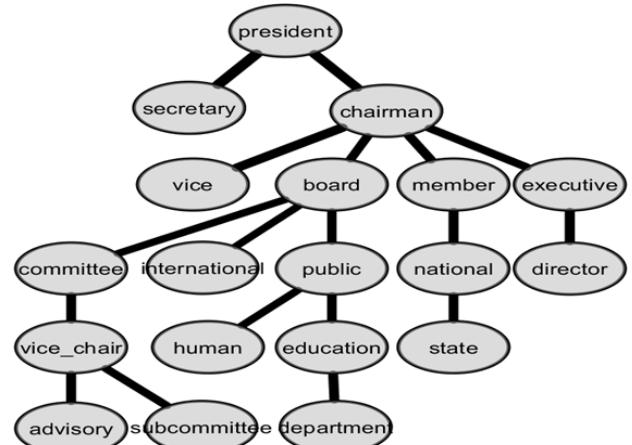
The second collocation measure we selected is the pointwise mutual information (PMI) between two words. In information theory, mutual information refers to the mutual information between two random variables rather than between two related events. It is a bad measure of dependence because it does not depend on the frequency. PMI is bad with sparse data, some words that only occur once, but appear together, score very high PMI. PMI is also bad with word dependence, when two words are perfectly dependent of each other, whenever one occurs; the other occurs, so the rarer the word is, the higher the PMI it gets. Bouma (2009) shows several normalised variants of PMI in order to make PMI less sensitive to a low frequency. We have selected the normalised pointwise mutual information ( $NPMI^3$ ) to the power of cube as it reduces the score for rare words.

$$PMI(x, y) = \log \frac{p(x, y)}{p(x)p(y)} \quad (12)$$

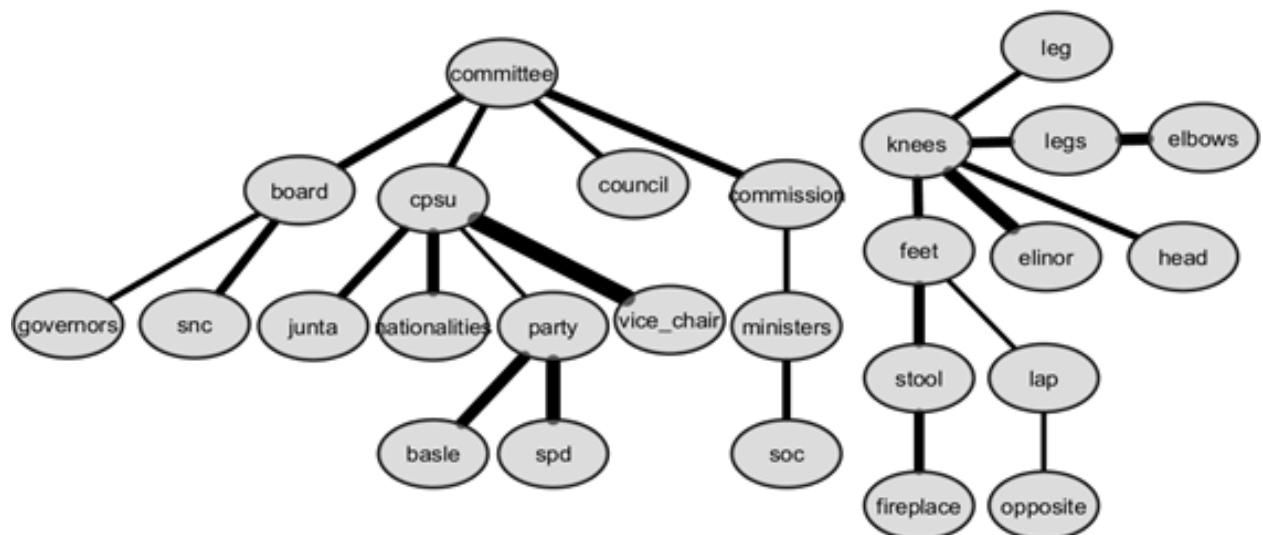
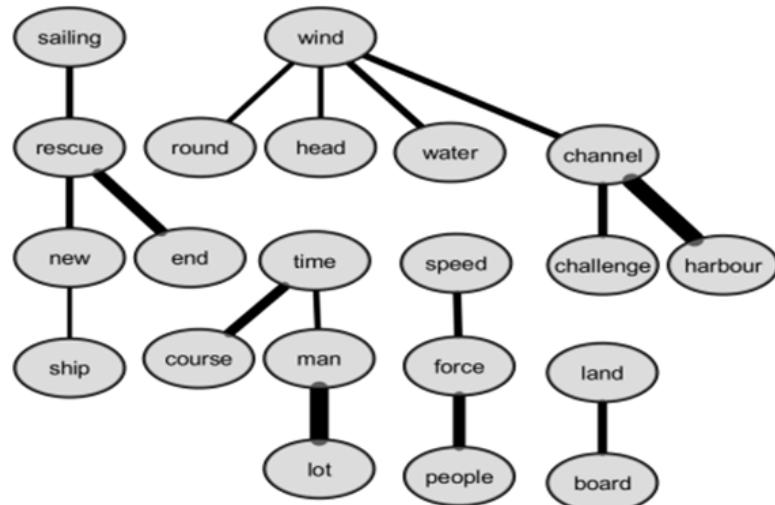
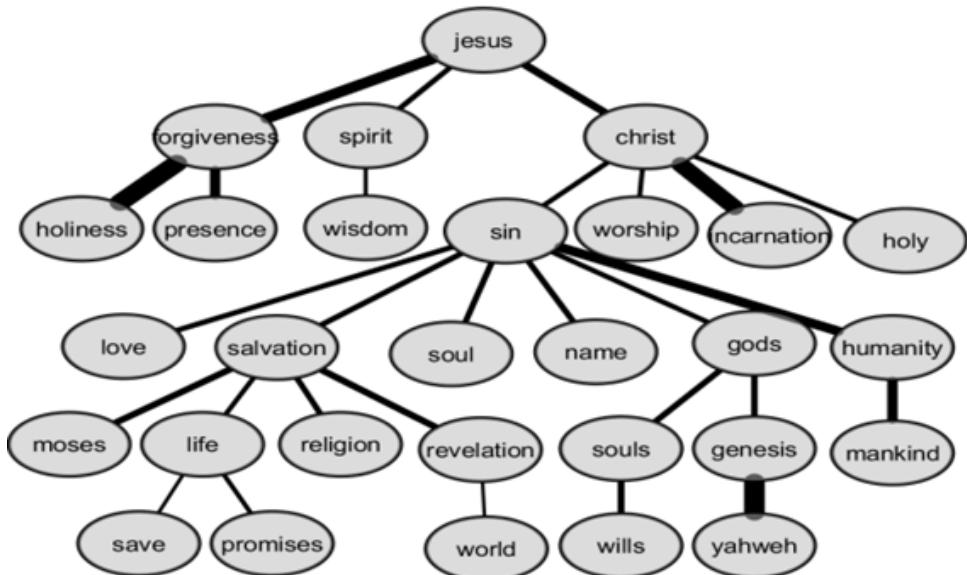
We calculate  $LLR$  and  $NPMI^3$  for each bi-gram, given a word we align the result from  $LLR$  and  $NPMI^3$  forming a feature vector of the given word. We use the top 1/3 of the characteristic vector as its collocation ratio is very low and is meaningless on the other 2/3.

We build a few graphs of words, one for each case (verb-verb, noun-noun, verb-noun, noun-verb, adverbverb, verb-adverb, adjective-noun, and noun-adjective) for BNC, en.wikipedia and Google n-gram.

**Figure 9** Word ‘chair’ extracted from en.wikipedia (noun-noun), it shows the largest cluster



For word sense disambiguation we do not cluster the whole graph, instead we get a subset, the subset which is related to the given word. The sub graph is constructed as follows: we get the feature vector of the given word and load it to the graph, then for each word in the feature vector we load its feature vector to the graph. In a similar approach to the work presented in Fern and Brodley (2004). Then, we cluster the sub graph using PaC to obtain clusters of senses of a given word.

**Figure 10** Word ‘chair’ extracted from BNC (noun-noun), it shows the largest clusters**Figure 11** Word ‘sail’ extracted from BNC (noun-verb) shows largest clusters**Figure 12** Word ‘god’ extracted from BNC (noun-noun) shows largest cluster

PaC produces clusters that represent word's senses, as shown in Figures 9 to 12 and Tables 11 and 12, the uJaccard similarity shows how well related words are. Edge thickness denotes the collocation measure given by uJaccard. We notice that each corpus shows different cluster as their contexts are different, as in the case of chair. The results are worthy to consider. Comparing with a base line such as WordNet or another repository will be cumbersome and not very productive because each corpus is completely different and from a different context (Agirre et al., 2004).

**Table 11** Senses/cluster for word apple, produced by PaC using en.wikipedia

macintosh, os, macos, mac, desktop
notebook, laptop, asus, baterry
wireless, usb, memory, remote
release, announce, calendar, make
store, product, user
imac, emac, intel, macbook
cranberry, pectin, cinnamon
juice, cider, martini
software, final, pro, power, hardware, advisor, video, quicktime
music, cinema, aria, garage band, book, deal, vacation, dvd, lossless, itunes, ipod, insight, ipods, archos
valley, spring, orchard, avery, tree, blossom, newton, loop, airport
support, core, developer, weblog, inspiration, iie, iiigs
powerbook, ibook, powermac, xserve

Note: The cluster related to the fruit is presented and other clusters are grouped by its senses such as music.

**Table 12** Senses for the word god, produced by PaC using Google bi-grams; it is clear that each cluster represents different aspects related to god

punish, forgive, almighty, forbid, instruct, spake, ordain
play, free, man, smile, soldier, live, tv
incarnate, bless, forsaken, sake, fear, manifest
told, knew, father, hear
send, save, word, rulesnet
damned, damn, awful, ente
mode, speed, command, channel
brought, bring, giveth, meant, reveal, bring, make, awfuls, gave
desire, set, choose, expect, require, dwell, change, raise, intend, accord
government, church, grant, design, exist, heal, lead

PaC produces clusters that represent word-senses, as shown in Tables 11 and 12, the uJaccard similarity shows how well related the words are. Google bi-grams corpus does not have any syntactic information or POS, we just use its frequency to get words collocation and then apply PaC. It is interesting to observe that even with lack of syntax information we get very meaningful word-senses.

## 7 Conclusions

A key assumption of most models of similarity is that a similarity relation is symmetric. The symmetry assumption is not universal, and it is not essential to all applications that use similarity. The need for asymmetric similarity is important and central in information retrieval and graph data networks. It can improve current methods and provide an alternative point of view.

We present a novel asymmetric similarity, uJaccard, where the similarity among  $A$  and  $B$  is not same to the similarity among  $B$  and  $C$ ,  $uJaccard(A, B) \neq uJaccard(B, A)$ ; this is based on the idea of paradigmatic association. In comparison to Tversky (1977), our approach uJaccard does not need a stimulus bias, whereas in the case of Tversky human judgement is needed.

We present a series of cases in which we confirm its usefulness and we validate uJaccard. We could extend uJaccard to include weights to improve the asymmetry. This is a task which we are working on Santisteban and Tejada-Cárcamo (2015c).

We have also shown how community detection can be approached using a paradigmatic analysis and uJaccard, in which the main idea is to find how similar two non-adjacent vertices are. Paradigmatic analysis is the process that identifies entities which are not related directly, but are similar because of their properties and their relatedness among other entities, similar to the regular equivalence. We introduce a new graph property, a paradigmatic property defined as the set of vertices that are similar if they have a regular equivalence and are interchangeable by substitution and transposition.

We also present a novel clustering algorithm PaC, which finds clusters with similar structural equivalence. The structural equivalence is found by using uJaccard. PaC is highly parallelised, incremental and its running time is  $O(n * avg(degree)^2)$ . Results show that PaC is able to find word-senses, as well as clusters on a ground-truth network.

In conclusion, the proposed uJaccard similarity coefficient is fundamental to several tasks, despite its simplicity. We propose a new paradigmatic graph property. Also, we proposed a PaC based on a uJaccard.

For future work, we would use uJaccard in different scenarios and fields and see where it fits best. We are working on a version of parallel PaC on a distributed system feed with a constant data streaming. Also, we are working on extending PaC in image segmentation. PaC is a general clustering algorithm and it is easy to implement in different scenarios and fields.

## Acknowledgements

This research was funded by ‘Fondo para la Innovación, Ciencia y Tecnología – Perú’ (FINCYT). We thank the Universidad Católica San Pablo for supporting this research.

## References

- Agirre, E., Alfonseca, E. and De Lacalle, O.L. (2004) ‘Approximating hierarchy-based similarity for wordnet nominal synsets using topic signatures’, in *Proceedings of GWC-04, 2nd Global WordNet Conference*, pp.15–22.
- Bouma, G. (2009) ‘Normalized (pointwise) mutual information in collocation extraction’, *Proceedings of GSCL*, pp.31–40.
- Bridge, D. (1998) ‘Defining and combining symmetric and asymmetric similarity measures’, in Smyth, B. and Cunningham, P. (Ed.): *Advances in Case-Based Reasoning (Procs. of the 4th European Workshop on Case-Based Reasoning), LNRAI*, Springer, Vol. 1488, pp.52–63.
- Consortium, B. et al. (2007) ‘The British national corpus, version 3 (bnc xml edition)’, *Distributed by Oxford University Computing Services PP the BNC Consortium*.
- Csardi, G. and Nepusz, T. (2006) ‘The igraph software package for complex network research’, *InterJournal, Complex Systems*, Vol. 1695, No. 5, pp.1–9.
- Danon, L., Diaz-Guilera, A., Duch, J. and Arenas, A. (2005) ‘Comparing community structure identification’, *Journal of Statistical Mechanics: Theory and Experiment*, Vol. 2005, No. 9, p.P09008.
- De Saussure, F., Baskin, W. and Meisel, P. (2011) *Course in General Linguistics*, Columbia University Press.
- Duda, R., Hart, P. and Stork, D. (2001) *Pattern Classification*, 2nd ed., John Wiley & Sons.
- Dunning, T. (1993) ‘Accurate methods for the statistics of surprise and coincidence’, *Computational Linguistics*, Vol. 19, No. 1, pp.61–74.
- Fano, R.M. and Hawkins, D. (1961) ‘Transmission of information: a statistical theory of communications’, *American Journal of Physics*, Vol. 29, No. 11, pp.793–794.
- Fellbaum, C. (2005) ‘Wordnet and wordnets’, in Brown, K. (Ed.): *Encyclopedia of Language and Linguistics*, pp.665–670.
- Fern, X.Z. and Brodley, C.E. (2004) ‘Solving cluster ensemble problems by bipartite graph partitioning’, in *Proceedings of the Twenty-First International Conference on Machine Learning*, ACM, p.36.
- Fortunato, S. (2010) ‘Community detection in graphs’, *Physics Reports*, Vol. 486, No. 3, pp.75–174.
- Fortunato, S. and Castellano, C. (2012) ‘Community structure in graphs’, in *Computational Complexity*, pp.490–512, Springer.
- Franz, A. and Brants, T. (2006) ‘All our n-gram are belong to you’, *Google Machine Translation Team*, p.20.
- Girvan, M. and Newman, M.E. (2002) ‘Community structure in social and biological networks’, *Proceedings of the National Academy of Sciences*, Vol. 99, No. 12, pp.7821–7826.
- Hochreiter, R. and Waldhauser, C. (2013) ‘Solving dynamic optimisation problems with revolutionary algorithms’, *International Journal of Innovative Computing and Applications*, Vol. 5, No. 3, pp.143–151.
- Holman, E.W. (1979) ‘Monotonic models for asymmetric proximities’, *Journal of Mathematical Psychology*, Vol. 20, No. 1, pp.1–15.
- Jimenez, S., Becerra, C. and Gelbukh, A. (2012) ‘Soft cardinality: a parameterized similarity function for text comparison’, in *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, Association for Computational Linguistics, Vol. 1, pp.449–453.
- Justeson, J.S. and Katz, S.M. (1995) ‘Technical terminology: some linguistic properties and an algorithm for identification in text’, *Natural Language Engineering*, Vol. 1, No. 1, pp.9–27.
- Knuth, D.E. (1993) ‘The Stanford graphbase: a platform for combinatorial algorithms’, in *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.41–43, Society for Industrial and Applied Mathematics.
- Lancichinetti, A. and Fortunato, S. (2009) ‘Community detection algorithms: a comparative analysis’, *Physical Review E*, Vol. 80, No. 5, p.056117.
- Lancichinetti, A., Fortunato, S. and Radicchi, F. (2008) ‘Benchmark graphs for testing community detection algorithms’, *Physical Review E*, Vol. 78, No. 4, p.046110.
- Lee, L., Pereira, F.C., Cardie, C. and Mooney, R. (1999) ‘Similarity-based models of word cooccurrence probabilities’, in *Machine Learning*, Citeseer.
- Lorrain, F. and White, H.C. (1971) ‘Structural equivalence of individuals in social networks’, *The Journal of Mathematical Sociology*, Vol. 1, No. 1, pp.49–80.
- Lusseau, D. (2003) ‘The emergent properties of a dolphin social network’, *Proceedings of the Royal Society of London B: Biological Sciences*, Vol. 270, Suppl. 2, p.S186–S188.
- McClosky, D., Charniak, E. and Johnson, M. (2006) ‘Reranking and self-training for parser adaptation’, in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp.337–344.
- Newman, M.E. (2006) ‘Finding community structure in networks using the eigenvectors of matrices’, *Physical Review E*, Vol. 74, No. 3, p.036104.
- Newman, M.E. and Girvan, M. (2004) ‘Finding and evaluating community structure in networks’, *Physical Review E*, Vol. 69, No. 2, p.026113.
- Nosofsky, R.M. (1991) ‘Stimulus bias, asymmetric similarity, and classification’, *Cognitive Psychology*, Vol. 23, No. 1, pp.94–140.
- Pantel, P.A. (2003) *Clustering by Committee*, PhD thesis, Citeseer.
- Pons, P. and Latapy, M. (2005) ‘Computing communities in large networks using random walks’, in *Computer and Information Sciences-ISCIS 2005*, pp.284–293, Springer.
- Real, E.M., do Carmo Nicoletti, M. and de Oliveira, O.L. (2014) ‘A closer look into sequential clustering algorithms and associated post-processing refinement strategies’, *International Journal of Innovative Computing and Applications*, Vol. 6, No. 1, pp.1–12.
- Sahlgren, M. (2006) *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic relations between Words in High-dimensional Vector Spaces*, Doctoral dissertation, Institutionen för lingvistik, Stockholm.
- Santisteban, J. and Tejada-Carcamo, J. (2015a) ‘Paradigmatic clustering for NLP’, *IEEE International Conference on Data Mining Workshops, ICDMW*.
- Santisteban, J. and Tejada-Carcamo, J. (2015b) ‘Unilateral Jaccard similarity coefficient’, *Graph Search and Beyond*, pp.23–27.
- Santisteban, J. and Tejada-Carcamo, J. (2015c) ‘Unilateral weighted Jaccard coefficient for NLP’, in *2015 14th Mexican International Conference on Artificial Intelligence (MICAI)*, pp.14–20.
- Shepard, R.N. (1974) ‘Representation of structure in similarity data: problems and prospects’, *Psychometrika*, Vol. 39, No. 4, pp.373–421.

- Tejada-Cárcamo, J., Calvo, H., Gelbukh, A. and Hara, K. (2010). ‘Unsupervised WSD by finding the predominant sense using context as a dynamic thesaurus’, *Journal of Computer Science and Technology*, Vol. 25, No. 5, pp.1030–1039.
- Tversky, A. (1977) ‘Features of similarity’, in *Psychological Review*, Vol. 84, No. 4, pp.327–352.
- Weeds, J. and Weir, D. (2005) ‘Co-occurrence retrieval: a flexible framework for lexical distributional similarity’, *Computational Linguistics*, Vol. 31, No. 4, pp.439–475.
- White, D.R. and Reitz, K.P. (1983) ‘Graph and semigroup homomorphisms on networks of relations’, *Social Networks*, Vol. 5, No. 2, pp.193–234.
- Yang, J. and Leskovec, J. (2015) ‘Defining and evaluating network communities based on ground-truth’, *Knowledge and Information Systems*, Vol. 42, No. 1, pp.181–213.
- Zachary, W.W. (1977) ‘An information flow model for conflict and fission in small groups’, *Journal of Anthropological Research*, pp.452–473.



## **Apendice E**

### **Artículo V**

---

*Santisteban, J. (Año 2012). “Método De Clusterización De Textos Basado En La Semántica Ontológica De La Lengua Inglesa”. Universidad Católica San Pablo, Revista De Investigación N° 3, Volúmen 3. ISSN Versión Impresa 2309-6683.*

# Método de clusterización de textos basado en la semántica ontológica de la lengua inglesa

M.Cs. Julio Omar Santisteban Pablo

*Máster de Ciencias en Internetworking por la University of Technology de Sidney, Australia.  
Profesional de Tecnología de Información, con experiencia laboral de más de 9 años en Australia.*

*Trabajó como Senior Analyst Programmer en Qantas Airways. Doctorando en Ciencias de la Computación por la Cátedra CONCYTEC de la Universidad Nacional de San Agustín, de Arequipa. Docente investigador de la UCSP.*



# Método de clusterización de textos basado en la semántica ontológica de la lengua inglesa

## Resumen

Muchas técnicas del procesamiento del lenguaje natural son enfocadas en procesos estadísticos y probabilísticos, convirtiendo un documento en una bolsa de palabras con sus respectivas frecuencias estadísticas. Algunos métodos como n-gramas, sinónimos y otros, son mejorados. Por otro lado, la presente investigación se enfoca en un método de análisis léxico y sintáctico con un enfoque semántico, también se presenta un nuevo método de cálculo de la similitud semántica entre oraciones, párrafos y documentos; asimismo, se propone una técnica de paráfrasis inversa. El objetivo de esta investigación es identificar similitudes semánticas.

En los últimos años se viene produciendo un gran interés en el análisis semántico de textos, incluso, existen procesos de incorporación de métodos semánticos en las páginas web. La visión detrás de estos métodos se fundamenta en la necesidad de poder identificar o reconocer lo que el autor de un documento trata de expresar.

Tres enfoques son los más usados para hacer un análisis semántico de los textos, documentos y páginas web. *The World Wide Web Consortium (W3C)* endosa la inclusión de etiquetas a las páginas web, etiquetas que incluyen atributos semánticos y ontológicos, indicando, de forma estructurada, sobre qué se está hablando. Con tecnologías tales como *Web Ontology Language (OWL)* el método llega a ser un metadato de la página web, de cierta forma un poco redundante, ya que se tendría que etiquetar o reescribir lo que el autor intenta expresar. Sin embargo, el mismo lenguaje inglés o español tiene su propia estructura formal.

Hoy en día, la utilización de métodos estadísticos, en conjunción con correlaciones entre palabras en uno o varios documentos o textos, es de uso común en la minería de datos. La clusterización, método inicialmente propuesto hace más de 35 años por Salton, Wong y Yang (1975), se basa en la frecuencia de palabras de un documento o texto, lo que permite agruparlos. Se debe considerar que los lenguajes son ricos para expresarse, que un autor podría expresar la misma idea utilizando distintas palabras, lo que conllevaría a tener una correlación baja entre las mismas, resultando palabras no clusterizadas.

Las técnicas actuales de clusterización y clasificación de textos usan métodos estadísticos, buscando o agrupando textos según la frecuencia de la ocurrencia de las palabras, dejando de lado el léxico, la sintaxis y la semántica (de quien se habla y sobre qué).

Los métodos léxicos para la resolución de la ambigüedad de sustantivos son usados como parte del núcleo de la presente investigación, siendo este un enfoque lingüístico y no meramente probabilístico. Asimismo, los métodos de paráfrasis y paráfrasis inversa, también constituyen parte del núcleo. Estos tres elementos permiten proponer un método novedoso de búsqueda de textos similares en un grafo ontológico, un grafo de la lengua inglesa, dado por la base de datos léxica de *WordNet*.

El objetivo de esta investigación es proponer un método novedoso de clusterización, enfocándose en la semántica ontológica, utilizando métodos léxicos y sintácticos para descubrir las similitudes entre textos.

El presente documento está estructurado de la siguiente forma, en el primer punto se presentarán los distintos métodos del procesamiento del lenguaje natural. Luego, se hablará del método de clusterización propuesto. En el siguiente punto, se presentarán dos experimentos utilizando dicho método y se expondrán sus resultados. Por último, se desarrollarán las conclusiones.

### *Métodos de procesamiento del lenguaje natural NLP*

*Método de Vectorial Probabilística.* Xia, Zong y Li (2011) argumentan que los métodos de representación de texto en un documento se consideran como una bolsa de palabras, la cual es representada por un vector que contiene todas las palabras que aparecen en el documento con sus respectivas frecuencias. Sin embargo, una parte de la información del documento original se descarta, se remueven las palabra de uso común (*stop words*); así también, el orden de las palabras se altera y las estructuras sintácticas se rompen.

Zhuge (2010) argumenta que, en un marco de inteligencia artificial y filosófico, la semántica se puede expresar en un modelo ontológico, manifestando funcionalidad y relación entre cosas/objetos y símbolos. Con un árbol sintáctico se procederá a la identificación de los hipónimos, hiperónimos y merónimos de cada grupo de palabras que conforman la oración. Estos serán extraídos de la base de datos léxica *WordNet* con el objetivo de identificar relaciones semánticas y léxicas basadas en dependencias, las cuales son utilizadas para procesarlas a través de SVM (*Support Vector Machine*) (Rink y Harabgiu, 2010). Se tiene que considerar que esta es una técnica probabilística que usa la sintaxis lingüística como extractor de datos.

El Análisis Semántico Latente (LSA) fue presentado por Deerwester, Dumais, Furnas, Landauer y Harshman (1990) como un método para la gestión y recuperación de documentos. La motivación principal detrás de LSA fue la utilización de la latente lingüística para comparar palabras con el objetivo de calcular la similitud entre documentos. La técnica matemática detrás de LSA también se conoce como Análisis de los Componentes Principales, una herramienta bien establecida en muchas áreas de aprendizaje estadístico. Los autores Seung y Lee (2000) y Landauer y Dumais (1997) sostienen que LSA es uno de los métodos con mejores resultados en la comparación de palabras. Se ha afirmado que los procesos de cognición/conocimiento humano tienen una función muy similar a la reducción dimensional.

Blei, Ng y Jordan (2003) introducen la Asignación Latente de Dirichlet (LDA) como una alternativa probabilística a LSA. LDA es un modelo de latencia variable y se refiere, frecuentemente, como un “modelo de temas”; su objetivo explícito es descubrir temas en los datos, donde se define un tema como una distribución multinomial sobre las palabras relacionadas. Una manera simple de entender lo que hace LDA, es agrupar las palabras en los temas y asignar a cada documento una distribución sobre esos temas. Griffiths, Steyvers y Tenenbaum (2007) demostraron que el modelo estándar de LDA tiene un buen rendimiento de desempeño en la predicción de los juicios humanos de la asociación de palabras, y supera a LSA en la tarea de sinonimia de opción múltiple.

La hipótesis de distribución ha dado la base teórica para la construcción de modelos del procesamiento del lenguaje natural. Utilizando la información de frecuencias prima el riesgo de que los tipos frecuentes de contexto puedan dominar la comparación de vectores, los tipos de contexto muy frecuentes pueden tener un alto grado de coocurrencia de cada palabra en un vector de palabras. Pero, ¿cuán frecuente es la coocurrencia?

*N-gramas.* Fuchun, Dale y Shaojun (2003) proponen un método de categorización de textos basado en un modelo de n-gramas, incorporando técnicas de Bayes Nativos, con el objetivo de identificar similitudes entre n-gramas. Los autores Pang, Lee y Vaithyanathan (2002) sustentan que las relaciones entre las palabras de una oración son usadas ampliamente, estas relaciones son obtenidas haciendo un análisis sintáctico de las oraciones y organizándolas en n-gramas de 2-gramas, 3-gramas o más. Sin embargo, su rendimiento es más bajo en comparación a los uni-gramas. Siendo estos métodos los que incorporan niveles estadísticos y probabilísticos en la categorización.

El modelo de análisis de contexto sintáctico utiliza identificadores lingüísticos para dependencias lingüísticas, tales como Objeto Directo, Objeto Indirecto, Modificador, Sujeto y Predicado. Tienen el objetivo de identificar n-gramas sintácticos usados en RASP (Briscoe y Carroll, 2002), MINIPAR (Lin, 1998) y *Stanford format*.

*Ambigüedad.* La Real Academia Española (2001a, 22.<sup>a</sup> ed.) define ambiguo como “Dicho especialmente del lenguaje: Que puede entenderse de varios modos o admitir distintas interpretaciones y dar, por consiguiente, motivo a dudas, incertidumbre o confusión”. Sea el caso de San Martín que podría confundirse entre el santo San Martín de Porres, el libertador José de San Martín y la Universidad de San Martín de Porres. La ambigüedad se da desde palabras sueltas a niveles más complejos como oraciones.

Costello, Veale y Dunne (2006) proponen un modelo de resolución de ambigüedad en palabras compuestas, asignando una puntuación de la Ocurrencia Relación Proporcional (PRO). La PRO es la identificación de palabras compuestas que tengan el mismo tipo de relación, estas palabras son generadas con los hiperónimos de las palabras cabecera y modificador, de la palabra compuesta; sus resultados obtuvieron 92 % de acierto en la identificación de la correcta relación.

En el Estudio de la expansión de consulta automática de Recuperación de Información de Carpineto y Romano (2012), se argumenta que las propiedades globales del lenguaje, tales como las relaciones morfológicas, léxicas, sintácticas y semánticas de palabras, son usadas para ampliar o redefinir los términos; utilizando, normalmente, los diccionarios, tesauros u otras fuentes similares de representación del conocimiento, tales como *WordNet*. Como las características de expansión suelen generarse independientemente del significado y de su contexto, por lo general, son más sensibles a palabras con sentidos ambiguos.

Tsang y Stevenson (2010) argumentan que la clasificación de documentos es similar al proceso de desambiguación de palabras. Su propuesta está basada en Red de Flujo (*Network Flow*), la cual es una ontología de *WordNet*, esta es cargada en un grafo que tiene pesos en los arcos, estos pesos son frecuencias de relación; la similitud se basa en identificar subgrafos similares con distribuciones también similares. En su propuesta, Tsang y Stevenson consideran un documento como un vector de palabras cargadas en un grafo; sus hallazgos fueron coherentes con una precisión de 68,7 %, logrado por Rennie y Rifkin (2001), usando una SVM (*Support Vector Machine*). Así también, Kozareva, Riloff y Hovy (2008) proponen el uso de algoritmos de grafos para identificar las verdaderas relaciones de pertenencia de clase en una colección, de forma automática, sin embargo, este grafo utiliza frecuencias en las aristas, el cual es construido utilizando *Bootstrap*.

*Paráfrasis.* La Real Academia Española (2001b, 22.ª ed.) define paráfrasis como 1. f.: “Explicación o interpretación amplificativa de un texto para ilustrarlo o hacerlo más claro o inteligible”; 2: “Traducción en verso en la cual se imita el original, sin verterlo con escrupulosa exactitud”. Madnani y Dorr (2010) definen paráfrasis como el principio de equivalencia semántica: una paráfrasis es una forma alternativa de construir discursos que expresan el mismo contenido semántico que la forma original. Las unidades léxicas que tienen el mismo significado, normalmente, se refieren como paráfrasis léxicas o, más comúnmente, sinónimos; sin embargo, parafraseando no puede limitarse estrictamente al concepto de sinonimia. Hay varias otras formas, tales como hipónimos, hiperónimos y otras. *WordNet* (Miller, 1995) define paráfrasis como “Expresar el mismo mensaje con

diferentes palabras”; “Reformulación con el propósito de aclaración”. Sin duda, el ser humano tiene la capacidad de elaborar versos/oraciones con características más genéricas o más específicas, y aun conservando el mismo significado semántico, el cual es un proceso cognoscitivo del hombre.

Harris (citado por Madnani y Dorr) hace unas observaciones empíricas “Las expresiones y frases no son producidas en forma arbitraria reuniendo los elementos de la lengua. De hecho, estos elementos suelen ocurrir solo en ciertas posiciones en relación con otros elementos” (p. 348). Harris indica que existe una relación entre el uso de las palabras y su significado, teniendo una correlación de uso entre palabras; y concluye su observación: “No todos los miembros de cada clase pueden ocurrir con todos los miembros de otra clase (sustantivos y adjetivos)” (p. 348). Harris denota que la paráfrasis de un verso/oración tiene un espacio finito, no siendo genérico.

*Partes de la oración.* Morante y Daelemans (2009) proponen la utilización de PoS, partes de la oración, como base fundamental de un clasificador, orientado al trabajo basado en oraciones y no solo en palabras o n-gramas. En su método de aprendizaje utilizan PoS como base, y para clasificar los textos biomédicos, con el objetivo de identificar señales de alto riesgo, utilizan métodos estadísticos y probabilísticas, tales como *K-Nearest Neighbor* (Cover y Hart, 1967), *Support Vector Machines* (SVM) y *Conditional Random Fileds* (CRFs).

El análisis léxico y sintáctico de las partes de la oración deberían ser indicadores significativos del sentido semántico. Investigaciones en la detección de la subjetividad revelan una alta correlación entre la presencia de adjetivos y la subjetividad de la oración (Hatzivassiloglou y Wiebe, 2000). Adicionalmente, las palabras más relacionadas semánticamente aparecen en el mismo documento, es muy probable que comparten el mismo sentido sobre un tema determinado, más aun, cada documento tiene sus propios rasgos en el estilo y la información sintáctica es una de las mejores medidas para la captura de estas divergencias (Gu, Zhu y Zhang, 2009).

*Extracción de relación.* Un área del procesamiento del lenguaje natural en la semántica computacional tiene como tarea identificar las relaciones entre las entidades expresadas en oraciones. La identificación del nivel de relación entre oraciones se llama Extracción de Relación, por ejemplo:

- Fumar causa cáncer.
- El cáncer debido al tabaco va en aumento.

Estas dos oraciones expresan una relación causal entre el tabaquismo y el cáncer.

En los procesos de SemEval se presentan mayormente tres tipos de técnicas para la clasificación de las relaciones semánticas entre nominales y extracción de relaciones.

- Las características contextuales se extraen del contexto de las entidades en la sentencia. Estas características corresponden a las palabras en el contexto, o pueden ser características estructuradas tales como subsecuenciales o fragmentos de dependencia.
- Las características léxicas describen las entidades, las cuales solo pueden corresponder al contenido de las entidades léxicas o también pueden codificar la información sobre entidades de distribución (Ó Saghdha y Copestake, 2009).
- Las características de relación utilizan las similitudes descubiertas por un sistema de detección que codifica la información contextual.

El análisis léxico, sintáctico y semántico es un área en la que existen varios enfoques pero los más usados son sinónimos, hipónimos, hiperónimos y merónimos (Yu et al., 2010), así como también métodos paradigmáticos y sintagmáticos (Tejada, 2009). El objetivo de estos métodos es encontrar palabras con el mismo significado, el resultado de este análisis semántico es incorporado en los métodos que usan frecuencias de ocurrencia de palabras, dando buenos resultados (Negri y Kouylekov, 2010). Por otro lado, no se hace un análisis léxico, sintáctico semántico de las oraciones que son base del sentido de ideas que un autor quiere expresar.

*Métodos de Similitud.* El método de medida de similitud de Wu y Palmer (1994) es uno de los métodos más comunes en taxonomías léxicas, calcula la profundidad de dos palabras en un árbol ontológico, por medio de la medición de la distancia a un hiperónimo común entre dos palabras distintas. Wu y Palmer no consideran relaciones tales como hipónimos o merónimos que expresan también una correlación entre palabras.

Algergawy, Mesiti, Nayak y Saake (2011) identifican medidas simples de similitud, como ocurre en el caso en que las medidas semánticas y sintácticas son incluidas para determinar el grado de similitud entre objetos y palabras; las medidas semánticas dependen, en gran parte, de los diccionarios externos o corpus creados manualmente, tales como *WordNet*.

Pedersen, Patwardhan y Michelizzi (2004) identificaron tres medidas de similitud generales: HSO (Hirst y St-Onge, 1998), Lesk (Banerjee y Pedersen, 2003) y el Vector (Pat-

wardhan, 2003). La medida HSO clasifica las relaciones de *WordNet* como de dirección, y después se establece la relación entre dos conceptos mediante la búsqueda de un camino más corto en un grafo. Las medidas de Lesk y Vectoriales incorporan la información de las glosas de *WordNet*. La medida Lesk encuentra coincidencias entre las glosas de dos conceptos y las glosas de palabras vinculadas. La medida Vectorial crea una matriz de coocurrencia de cada palabra dada de las glosas en *WordNet*, siendo este el vector promedio de coocurrencia.

Los autores Banerjee y Pedersen extendieron el trabajo de Michael Lesk usando relaciones de palabras, tales como hiperónimos, para extraer las glosas con sentidos similares, su trabajo dio el mejor resultado en el SENSEVAL-2. Basado en el trabajo de Lesk (1986) y Banerjee y Pedersen podemos concluir que el uso de la comparación de glosas para identificar el sentido de las palabras en una oración, es ventajoso desde el punto de vista que no se utiliza algún tipo de sintaxis del lenguaje, más aun podrá proveer mejores resultados incorporando aspectos lingüísticos.

Negri y Kouylekov (2010) utilizan un solo tipo de relación, “IS A” (es un) derivada de las formas hipónimos y merónimos para encontrar relaciones entre nominales de una oración. Adicionalmente, los autores mencionan el uso de hipónimos e hiperónimos para encontrar una relación transitiva, mediante la cual se podría encontrar una relación entre palabras tales como *manada* y *animal*.

*WordNet*. Es una gran base de datos léxica de inglés. Sustantivos, verbos, adjetivos y adverbios se agrupan en conjuntos de sinónimos (synsets cognitivas), cada uno expresando un concepto distinto. Los synsets están vinculados entre sí por medio de las relaciones conceptuales, semánticas y léxicas. Lo cual resulta en una red de las palabras y los conceptos relacionados de manera significativa (Miller).

Márquez, Carreras, Litkowski y Stevenson (2008) argumentan que existen pocas investigaciones en las relaciones semánticas y el uso de corpus del conocimiento, tales como *WordNet* u otros, los cuales podrían proporcionar mejoras en NLP y dejar de lado los modelos cuantitativos. Adicionalmente, Clarke y Lapata (2010) argumentan que en la resolución de la correferencia o la identificación de las relaciones gramaticales, las cadenas léxicas pueden ser más robustas utilizando un diccionario léxico como *WordNet*, el cual provee relaciones léxicas, por ejemplo, hipónimia, sinonimia u otros.

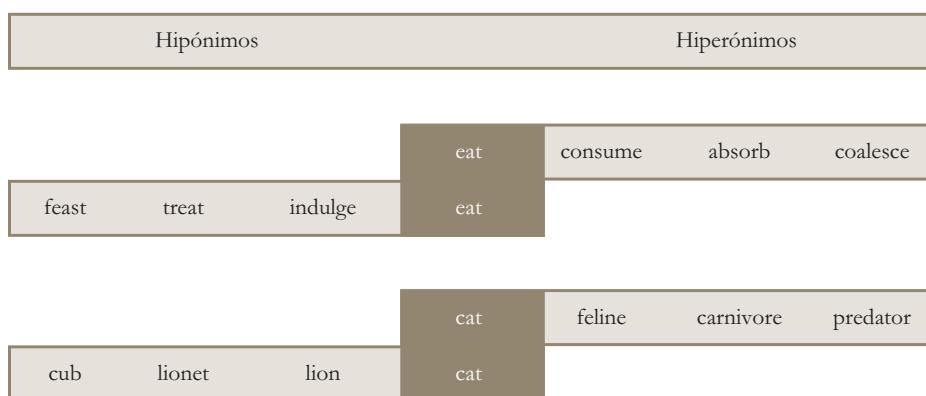
La correlación entre oraciones no solo se mide al utilizar hipónimos, hiperónimos o merónimos sino también con hiperónimo, hipónimo, ejemplo-hiperónimo, ejemplo-hi-

pónimo, holonym parte, merónimo parte, miembro holonym, miembro merónimo, sustancia holonym, sustancia merónimo, implica, causa, antónimo, similares, también, atributo, verbo grupo, participio, pertainym, derivación, dominio de la categoría, categoría de miembro de dominio, dominio de la región, dominio de la región miembros, dominio de uso, dominio de uso de los miembros, de dominio y miembros, los cuales son relaciones ontológicas en la base de datos léxica *WordNet* (Miller). Resultante en un red de palabras, extrapolándolo en un grafo ontológico.

### *Método de clusterización de textos*

Los autores Dragut, Fang, Sistla, Yu y Meng (2009) argumentan que se pueden utilizar sinónimos e hiperónimos para extraer relaciones semánticas entre etiquetas en un dominio contextual, argumentan que la relación de hiperonimia es transitiva, asimétrica e irreflexiva y de sinonimia equivalente. Sus pruebas alcanzan un 95 % de precisión en contexto de etiquetas y palabras semánticas. La propuesta de Costello et al. (2006) de utilizar hiperónimos con el fin de generar nuevas palabras compuestas que tengan el mismo tipo de relación pero mas genéricas, sustenta la utilización de hiperónimos como medio de extrapolación del significado semántico. Más aun, Wu y Palmer (1994) proponen la utilización de relaciones hipónimos, hiperónimos o merónimos con el objetivo de encontrar relaciones semánticas entre palabras; así también concuerdan Banerjee y Pedersen; Costello et al.; Negri y Kouylekov; Madnani y Dorr; Clarke y Lapata; Yu et al. (2010); Zhuge; Algergawy et al. (2011).

Los hipónimos e hiperónimos nos permiten extraer palabras más genéricas y más específicas, respectivamente. En un análisis empírico del ser humano podemos ver que al construir oraciones se expresan de forma genérica y, en muchos casos, de forma más específica, esto es un proceso de parafrasear las ideas. Dado este enfoque podemos concluir que es razonable usar hipónimos e hiperónimos con el objetivo de buscar ideas semánticamente iguales.



Algunos hipónimos e hiperónimos del verbo *eat* y el sustantivo *cat*.

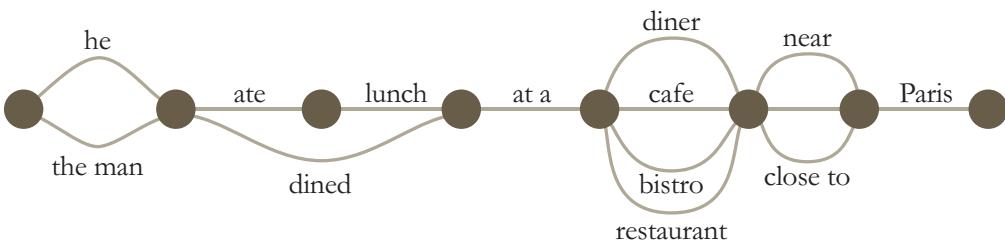
Se puede observar que usando hipónimos e hiperónimos es posible obtener palabras (verbo y sustantivo) más genéricas y más específicas.

La utilización de grafos para encontrar hipónimos, hiperónimos, merónimos y demás relaciones será una parte del núcleo de la implementación de esta investigación, los grafos representarán las relaciones ontológicas de la lengua, el grafo será una extrapolación de las relaciones léxico sintácticas de *WordNet* (Miller). *WordNet* será la fuente primaria de este grafo ontológico, las relaciones se encontrarán utilizando el algoritmo de búsqueda en grafos Dijkstra, que resuelve el problema de la ruta más corta (Dijkstra, 1959), y BFS (*Breadth First Search*), los cuales identificarán las relaciones más cercanas entre dos palabras.

Tsang y Stevenson (2010) argumentan que la clasificación de documentos es similar al proceso de resolución de ambigüedad de palabras. El nivel de ambigüedad es más alto en textos que en una palabra. Hatzivassiloglou y Wiebe argumentan que existe una alta correlación entre la presencia de adjetivos y la subjetividad de la oración, siendo un indicador significativo del sentido; punto que disminuye la ambigüedad en las oraciones, puesto que existe un contexto en la oración y es dada por el objeto directo, objeto indirecto, modificador, sujeto y predicado.

Paráfrasis inversa es parte del núcleo de la clusterización propuesta, en este caso se realiza un proceso inverso al del paráfrasis. A diferencia de generar versos/oraciones con diferentes palabras pero con el mismo significado, la propuesta de esta investigación es dada por la identificación de versos/oraciones que tengan el mismo significado semántico mas no usen las mismas palabras o, en algunos casos, no usen la misma estructura gramatical, lo cual es el proceso inverso al de paráfrasis. Este proceso se logrará al encontrar oraciones con el mismo sentido semántico, significado.

El proceso de la paráfrasis inversa se alcanzará por medio de la descomposición de las partes de una oración dada, tales como sujeto, verbo, adjetivos, adverbios de la oración; con estas palabras etiquetadas podremos utilizar un método de extracción de hipónimos e hiperónimos (palabras más genéricas y más específicas). Se extraerán solo aquellos hipónimos e hiperónimos dentro de un grado de profundidad no mayor a 4. Se debe considerar que a este punto el grado de ambigüedad es alto ya que la extracción de hipónimos e hiperónimos no distingue entre los distintos sentidos de cada palabra. Extraídos los hipónimos e hiperónimos podremos construir oraciones, las cuales serán parafraseadas de la original.

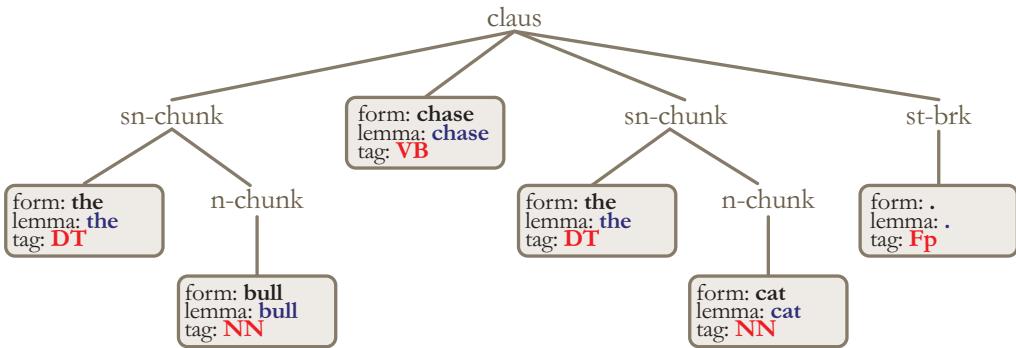


Fuente: Madnani y Dorr, Figura 6.

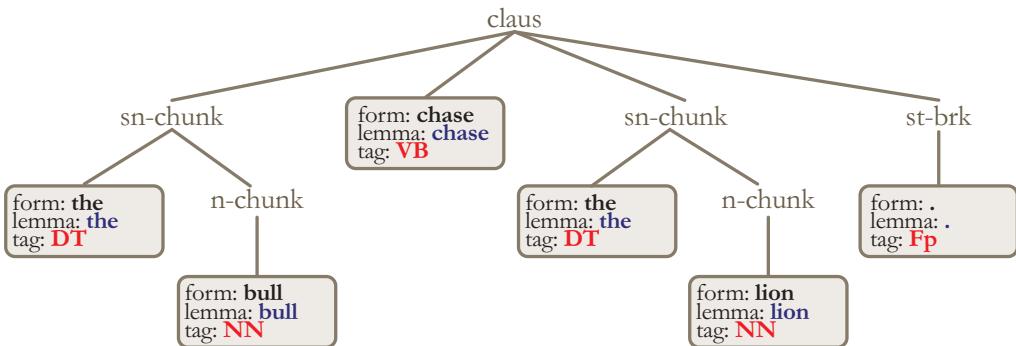
Se puede observar que se puede generar un número de oraciones remplazando algunas palabras en la oración.

La construcción de las oraciones será hecha en un grafo ontológico, el cual está definido por palabras como vértices y relaciones como aristas; las relaciones serán extrapoladas usando los 22 tipos de relaciones en *WordNet*. Cada paráfrasis y cada párrafo de un texto serán dados por un subgrafo(s). El objetivo de la paráfrasis inversa es encontrar los grafos similares o iguales, encontrado estos se podrá afirmar que las oraciones, origen de estas paráfrasis, son semánticamente similares.

La clusterización se da en la agrupación de los subgrafos resultantes de la paráfrasis inversa, estos subgrafos se agrupan según su mismo significado semántico. El proceso de clusterización semántica se basa en el análisis sintáctico y léxico de las oraciones. Las oraciones de textos, que serán analizadas por el utilitario *FreeLing 2.2* (Padró, Collado, Reese, Lloberes y Castellón, 2010), proporcionan un analizador de idiomas morfológico, generando un árbol sintáctico de las oraciones, etiquetando las partes de la oración (POS), en el cual se identifican sustantivos, verbos, adjetivos, adverbios y otros. Una vez obtenidas las PoS se someterán al grafo ontológico con el fin de encontrar oraciones parafraseadas con el mismo significado semántico. Desde un enfoque de paráfrasis inversa, *FreeLing* también proporciona el sentido de las palabras que podrán mejorar el método, este usa técnicas como MFS (Sentido Más Frecuente) y UKB (Gráfico de Textos Basado en Desambiguación de Sentidos y Semejanzas).



Fuente: Generado con *Freeling 2.2*, “*the bull chase the lion*”.



Fuente: Generado con *Freeling 2.2*, ”*the bull chase the cat*”.

La oración “*the bull chase the cat*” se relaciona más a “*the bull chase the lion*” que a “*the bull chase the lionet*”, ya que la relación entre *cat* a *lion* es de 1 y la relación de *cat* a *lionet* es de 2. Así también, podemos decir que “*the bull chase the cat*” se relaciona más a “*the cattle chase the lionet*” que a “*the jax chase the lion*” ya que la relación de *bull* a *cattle* que es de 1, más la relación de *cat* a *lionet* que es de 2, la relación de las dos oraciones es de 3, en comparación, la relación entre *bull* a *jax* es de 4 y la relación entre *cat* a *lion* es de 1, la relación de las dos oraciones es de 5.

La relación entre las oraciones será la suma de las relaciones de sus palabras, manteniendo el orden de la oración (orden del sujeto, verbo, adjetivo y sustantivo), las oraciones más relacionadas serán aquellas que tengan el mayor grado de relación (ruta más corta en el grafo).

La relación entre los párrafos será la suma de las relaciones de sus oraciones, manteniendo el orden del párrafo, los párrafos más relacionados serán aquellos que tengan el mayor grado de relación (ruta más corta entre subgrafo).

La relación entre textos/documentos no se da entre texto/documento a texto/documento ya que podrían producir un alto grado de ambigüedad, más aun, en un texto/documento podremos encontrar un número de temas que causarían mayor ambigüedad. La similitud entre textos/documentos se da por el grado de similitud de sus párrafos.

La clusterización está dada por los subgrafos (párrafos) cúmulos en el grafo ontológico, estos subgrafos son el número de subgrafos adyacentes o próximos a un subgrafo dado; se podría mejorar el algoritmo extrayendo subgrafos similares y no tan solo subgrafos adyacentes. La clusterización es un término inglés, el término español es racimo. La Real Academia Española (2001c, 22.<sup>a</sup> ed.) define racimo en su punto 3 como: “Conjunto de cosas menudas dispuestas con alguna semejanza de racimo”. Los subgrafos cúmulos se agrupan en el grafo ontológico según su significado semántico o tema(s), produciendo cúmulos de párrafos.

### **Generación de un grafo ontológico:**

- Extraer las palabras de *WordNet*
- Extraer los sentidos de cada palabra
- Extraer los 22 tipos de relaciones
- Generar relaciones “ISA” de la palabra base a la palabra con su sentido
- Extraer las palabras con sus respectivas relaciones
- Extraer las palabras con su sentido y sus respectivas relaciones
- Preparar una estructura de datos, un grafo bidireccional
- Cargar las palabras como vértices en el grafo
- Cargar las palabras con sus sentidos como vértices en el grafo
- Cargar las relaciones como aristas en el grafo
- Implementar el método de recorrido Dijkstra para el grafo
- Implementar el método de recorrido BFS (*Breadth First Search*) para el grafo

### **Cargar el grafo con los textos/documentos:**

Dado una serie de textos/documentos (en formato texto).

Para cada texto:

- Se le asignará un identificador único DocID
- Se extraerán los párrafos

Para cada párrafo:

- Se le asignará un identificador único ParID
- Se extraerán las oraciones

Para cada oración (O):

- Se le asignará un identificador único OraID
- Se deberán encontrar las partes de la oración (PoS) utilizando *FreeLing* u otro analizador lingüístico
- En el resultado del análisis se identificará como mínimo 1 verbo y 1 sustantivo o más
  - => O<sub>verbo</sub> O<sub>adverbio</sub> O<sub>sustantivo</sub> O<sub>adjetivo</sub>
- El analizador identificará el sentido de la palabra usada en la oración
  - => O<sub>verbo</sub>Sentido<sup>X</sup> O<sub>adverbio</sub>Sentido<sup>X</sup>
  - O<sub>sustantivo</sub>Sentido<sup>X</sup>O<sub>adjetivo</sub>Sentido<sup>X</sup>
  - //Este dependerá del analizador
- Cargar PoS+SentidoX en el grafo como una ocurrencia del vértice
- Cada ocurrencia está identificada por DocID, ParID, OraID, Posad
- Fin para cada texto
- Fin para cada párrafo
- Fin para cada oración

### **Identificar oraciones similares:**

Dado el grafo cargado con *WordNet*.

Dado el grafo cargado con los textos/documentos, en la forma de ocurrencias:

Para cada PoS:

- Utilizando BFS encontrar ocurrencias en una profundidad de 5
- Registrar las ocurrencias encontradas
- Fin para cada Pos

Dadas las ocurrencias de cada oración:

- Identificar las oraciones en las ocurrencias
- Identificar las oraciones de interés, solo aquellas que tengan 2 o más ocurrencias
- Descartar las otras ocurrencias

Para cada oración de interés:

- Calcular el grado de similitud con la oración Od
- Registrar el grado de similitud
- Fin para cada oración de interés
- Fin para cada oración [ DocID, ParID, OraID ] Od

### **Calcular el grado de similitud de oraciones:**

Dada una oración base O<sub>d</sub>.

Dada una oración de interés, la cual incluye más de 2 ocurrencias O<sub>o</sub>.

Para cada PoS en la Oración O<sub>d</sub>:

Si  $O_{d\text{-Pos}}$  tiene una o más ocurrencias en la oración  $O_o$ :

- Identificar la ocurrencia con menor grado de profundidad
- Calcular peso de similitud:

Si grado es de 0 => similitud es 100 %

Si grado es de 5 => similitud es 50 %

Si grado es de 10 => similitud es 0 %

// método simplista o porcentual

Registrar el grado de similitud  $O_{d\text{-Pos}}\text{-}%$

Si  $O_d\text{-Pos}$  tiene una o más ocurrencia en la oración  $O_o$ :

- Calcular la similitud de  $O_d$  en relación a  $O_o$

Dado que la sumatoria de los PoS de  $O_d$  \* 100 % =  $T_1$  = 100 % de similitud.

Para cada PoS de  $O_d$ :

- Si  $O_d\text{-Pos}\text{-}%$  No tiene ocurrencia: Acumular 0 % en  $S_1$
- De lo contrario: Acumular  $O_d\text{-Pos}\text{-}%$  en  $S_1$
- Fin para cada PoS de  $O_d$
- Retornar  $S_1$ ;  $S_1 \leq T_1$
- Fin para cada PoS en la Oración  $O_d$

### Identificar párrafos similares:

Dada una serie de oraciones Serie- $O_d$

Dados los grados de similitud de cada oración  $O_d$  con oraciones  $O_{o-n}$

Para cada párrafo [ DocID, ParID]  $P_d$

Para cada oración en el párrafo

- Basado en la similitud de oraciones
- Utilizando BFS para encontrar ocurrencias en una profundidad 5
- BFS buscará subgrafos
- Dos subgrafos son relacionados si uno de sus PoS son cercanos en profundidad  
// se podría mejorar encontrando el centroide (verbo y sustantivo)
- Registrar las ocurrencias encontradas
- Fin Para cada oración en el párrafo

Dada las ocurrencias:

- Identificar los párrafos (subgrafos) en las ocurrencias
- Identificar los párrafos de interés, solo aquellos que tengan 2 o más ocurrencias
- Descartar las otras ocurrencias

Para cada párrafo de interés:

- Calcular el grado de similitud con el párrafos  $P_d$
- Registrar el grado de similitud
- Fin para cada párrafo de interés

Para cada párrafo [ DocID, ParID]  $P_d$

- Calcular el grado de similitud de párrafos:

El cálculo es el mismo que el “Cálculo del grado de similitud de oraciones”, excepto que son párrafos.

- Calcular el grado de similitud de textos:

Se da en base a párrafos

No se calcula el texto en su totalidad, puesto que sería muy ambiguo

Un documento es similar a otro por el número de párrafos similares

El grado de similitud de un documento a otro es la suma del grado de similitud de sus párrafos sobre el número de párrafos similares.

### Calcular la clusterización:

- El grafo dado está representando una ontología lingüística
- El grafo dado está cargado con las similitudes de oraciones
- El grafo dado está cargado con las similitudes de párrafos (subgrafos)

Para cada subgrafo:

- Calcular el número de subgrafos ubicados en las proximidades  
// se podría mejorar el algoritmo ubicando subgrafos similares y sus próximos
- Fin para cada subgrafo
- Identificar los subgrafos cúmulos
- Subgrafos con un alto número de subgrafos alrededor de este  
// se podría mejorar el algoritmo extrayendo subgrafos similares cúmulos
- Un *cluster* está dado por los subgrafos cúmulo

### *Experimento*

Los experimentos ejecutados fueron dos, el primero fue hecho con el objetivo de validar el proceso de Paráfrasis Inversa que es parte esencial del núcleo de la propuesta. El segundo experimento fue comparativo, se utilizó un corpus desarrollado por Rennie y Rifkin, con el objetivo de poder tener un modo para comparar y validar los resultados.

*Experimento 1.* El objetivo del experimento fue de validar la Paráfrasis Inversa con el fin de procesar la extracción de relaciones entre oraciones. En este experimento solo se procesaron oraciones sueltas, este método permite encontrar oraciones que tengan el mismo significado. Lo que se busca es poder encontrar glosas semánticamente similares pero que correspondan a distintos conceptos, se observarán las relaciones y grado de similitud entre glosas y conceptos.

Las glosas se obtendrán de *WordNet*, con un total de 147 306 conceptos y un total de 117 659 glosas y 48 356 oraciones ejemplo; se debe considerar que cada palabra puede tener más de una glosa debido al sentido de la misma. Solo se tomaron 1000 glosas de forma aleatoria.

Pasos en el experimento:

- Generación de un grafo ontológico
- Cargar el grafo con las oraciones
- Identificar oraciones similares
- Calcular el grado de similitud de oraciones

Validación:

- Extraer los conceptos referidos a las glosas usadas
- Calcular similitud entre conceptos según HSO (Hirst y St-Onge).
- Comparar grado de similitud entre conceptos y sus respectivas glosas

## *Resultados*

Los resultados fueron razonables, teniendo 764 glosas con un grado de similitud mayor a 66 %, se debe notar que no existen similitudes en el rango 91 %-100 % ya que esto indicaría que la glosa es la misma. Existe una correlación con los conceptos de las glosas, de las 764 glosas, solo 592 conceptos (77.5 %) tienen un grado similar de similitud. Esto indica un nivel de relevancia entre similitud de las glosas y la similitud de sus conceptos. Son mínimos los conceptos y glosas que tienen el mismo grado de similitud.

1000 Glosas						
Similitud entre glosas	0 % - 33 %		33 % - 66 %		66 % - 100 %	
	113		123		764	
Similitud entre conceptos	66 % - 100 %	17	66 % - 100 %	17	66 % - 100 %	17
	33 % - 66 %	9	33 % - 66 %	9	33 % - 66 %	9
	0 % - 33 %	87	0 % - 33 %	87	0 % - 33 %	87

Tabla 1

Resultados de similitud entre glosas con sus palabras concepto

*Experimento 2.* El objetivo del segundo experimento, fue la clusterización de textos, los datos de nuestro corpus de escritos serán los del corpus preparado por Rennie y Rifkin para su investigación “Improving multiclass text classification with the Support Vector Machine”; Tsang y Stevenson también utilizaron el mismo corpus en su trabajo “A graph-theoretic framework for semantic distance”, los cuales servirán como base de una comparación.

El corpus es una colección de artículos de 20 diferentes grupos de noticias de Usenet, cada grupo de noticias corresponde a un tema, los artículos pueden ser clasificados con la etiqueta de grupo de noticias. En la colección se han eliminado todas las copias (mensajes cruzados), dando como resultado 18 828 artículos. Los artículos son distribuidos equitativamente entre los 20 grupos de noticias, aproximadamente. Las palabras vacías y los encabezados de artículos se han retirado.

Solo se seleccionaron 500 artículos correspondientes a 10 temas de noticias, se utilizó *Unix shell scripting* para formatear y eliminar data no necesaria para la clusterización. Se extrajeron párrafos con una longitud mayor a 400 caracteres, cada párrafo se dividió en oraciones, se utilizaron los signos de puntuación para separar las oraciones, tales como punto final, punto y coma, comillas dobles y las conjunciones *and, or, but, for, so, yet, nor*.

Pasos en el experimento:

- Generación de un grafo ontológico
- Cargar el grafo con los artículos
- Identificar oraciones similares
- Calcular el grado de similitud de oraciones
- Identificar párrafos similares
- Calcular el grado de similitud de párrafos
- Calcular el grado de similitud de textos
- Calcular la clusterización

Validación:

- Identificar los *clusters* generados
- Comparar los *clusters* con los temas de los grupos de noticias

## *Resultados*

No es claro establecer una comparación entre el presente método de clusterización y el método de clasificación trabajado por Rennie y Rifkin y Tsang y Stevenson, ya que los métodos de clasificación usados presumen la preexistencia de clases, de forma contraria a los método de clusterización, en que los elementos se ordenan en grupos según la información que los datos representan.

El presente método puede compararse al trabajo de los autores nombrados en el párrafo anterior, en el número de *clusters* generados, los cuales reflejan temas de clases de noticias preexistentes; por otro lado, la identificación de cúmulos en el grafo proporciona un número variable de *clusters* debido a la profundidad o expansión de los cúmulos dentro del grafo, obteniendo cúmulos con una dispersión baja o con una dispersión alta, para este experimento se fijó una profundidad de 5.

Clusters	Artículos	Temas							
		Atheism	Graphics	Pc.hardware	Windows.x	Forsale	Autos	Crypt	Med
1	3	x				x	x		
2	19	x					x		
3	5	x						x	
4	36	x							x
5	6	x							x
6	11	x							
7	1	x							
8	5	x							
9	31		x		x			x	
10	19		x		x				
11	26		x		x				
12	20		x			x	x		
13	28		x						x
14	25			x	x			x	
15	15			x	x				
16	36			x					x
17	26			x					
18	13			x					
19	31					x	x		
20	18					x	x		
21	12					x			x
22	11					x			x
23	17					x			
24	10					x			
25	4					x			
26	16						x		
27	1						x		
28	1							x	
29	23							x	
30	8							x	
31	7							x	x
32	9							x	
33	7							x	

Tabla 2  
Resultados de clusterización

En la Tabla 2 se observan 33 *clusters* obtenidos. En la columna 2 se muestran los artículos con mayor presencia en el *cluster* correspondientes a temas de noticias, se debe considerar que los *clusters* podrían incluir otros artículos si se extienden los cúmulos obtenidos, *clusters* con mayor dispersión. En las columnas 3 a la 12 se muestra la pertenencia de los artículos en los *clusters* a los temas de noticia. Se puede observar que, a pesar de tener un mayor número de clases (*clusters*), estos pertenecen a un tema de noticia, así también un *cluster* puede contener más de un artículo perteneciente a más de un tema de noticia.

### ***Conclusión***

En esta investigación, se presenta un nuevo método de clusterización basado en la semántica de la lengua inglesa, la idea principal es utilizar técnicas de análisis léxicos y sintácticos para clusterizar textos o documentos. Dejando de lado los métodos estadísticos y probabilísticos, la técnica de Paráfrasis Inversa es introducida como mecanismo para identificar oraciones similares, siendo esta una aportación de esta investigación. El uso de análisis léxicos y sintácticos proporciona un enfoque cognoscitivo en el procesamiento del lenguaje natural, clusterizando ideas y no tan solo vectores característicos. Cabe mencionar que sin un adecuado thesaurus, como *WordNet*, este clusterizador no podría darse, así también las limitaciones encontradas en el clusterizador son un reflejo de la cobertura que tiene *WordNet*, siendo una futura investigación la expansión de este *thesaurus*.

Las aplicaciones de este método no solo se limitan a la clusterización sino también a varias áreas del procesamiento del lenguaje natural, tales como resúmenes, sistemas de detección de plagios, indexadores de páginas web, detección de sistemas de opinión, entre otras.

En este estudio se presentan las bases del método clusterización semántica, dejando algunas mejoras por hacer en una investigación futura. Los resultados de los experimentos son aceptables, no obstante, más adelante se podría tener una validación más detallada. Aunque solo se usó una medida de profundidad en el grafo, no se descarta utilizar otras medidas de similitud en trabajos futuros.

## Referencias

- Algergawy, A., Mesiti, M., Nayak, R. y Saake, G. (2011). XML data clustering: An overview. *ACM Comput. Surv.*, 43( 4), pp. 1-41.
- Banerjee, S. y Pedersen, T. (2003). Extended gloss overlaps as a measure of semantic relatedness. *Proceedings of the 18th international joint conference on Artificial intelligence*, Acapulco, México.
- Blei, D. M., Ng, A. Y. y Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, pp. 993-1022.
- Briscoe, T. y Carroll, J. (2002). Robust accurate statistical annotation of general text. *Proceedings of the Third International Conference on Language Resources and Evaluation*, Las Palmas, España, pp. 1499-1504.
- Carpinetto, C. y Romano, G. (2012). A Survey of Automatic Query Expansion in Information Retrieval. *ACM Computing Surveys*, 44(1), pp. 1-50.
- Clarke, J. y Lapata, M. (2010). Discourse constraints for document compression', *Computational Linguistics*, 36(3), pp. 411-441.
- Costello, F. J., Veale, T. y Dunne, S. (2006). Using WordNet to automatically deduce relations between words in noun-noun compounds. *Proceedings of the COLING/ACL on Main conference poster sessions*, Sydney, Australia.
- Cover, T. y Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1), pp. 21-27.
- Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. y Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41, pp. 391-407.
- Dijkstra, E. W. (1959). A Note on Two Problems in Connection with Graphs. *Numerical Mathematics*, 1, pp. 269-271.
- Dragut, E., Fang, F., Sistla, P., Yu, C. y Meng, W. (2009). Stop word and related problems in web interface integration. *Proceedings of the VLDB Endowment*, 2(1), pp. 349-360.
- Fuchun, P., Dale, S. y Shaojun, W. (2003). Language and task independent text categorization with simple language models. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 1.
- Gu, P., Zhu, Q. y Zhang, C. (2009). A multi-view approach to semi-supervised document classification with incremental Naive Bayes. *Computers & Mathematics with Applications*, 57(6), pp. 1030-1036.
- Griffiths, T. L., Steyvers, M. y Tenenbaum, J. B. T. (2007). Topics in Semantic Representation. *Psychological Review*, 114(2), pp. 211-244.
- Hatzivassiloglou, V. y Wiebe, J. M. (2000). Effects of adjective orientation and gradability

- on sentence subjectivity. *Proceedings of the 18th conference on Computational linguistics*, Saarbrücken, Germany, 1.
- Hirst, G. y St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropism. En Fellbaum, C. (Ed.), *WordNet: An electronic lexical database* (pp. 305-332). Cambridge, MA: The MIT Press.
- Kozareva, Z., Riloff, E. y Hovy, E. (2008). Semantic class learning from the Web with hyponym pattern linkage graphs. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, Ohio, pp. 1-9.
- Landauer, T. K. y Dumais, S. T. (1997). A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, (104), pp. 211-240.
- Lin, D. (1998). Dependency-based evaluation of MINIPAR. *Proceedings of the Workshop on Evaluation of Parsing Systems*, Granada, España.
- Márquez, L., Carreras, X., Litkowski, K. C. y Stevenson, S. (2008). Semantic role labeling: an introduction to the special issue. *Computational Linguistics*, 34(2), pp. 145-159.
- Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. *Proceedings of the 5th annual international conference on Systems documentation*, Toronto, Ontario, Canada.
- Miller, G. A. (1995). Word Net: A Lexical Database for English. *Communications of the ACM*, 38(11), pp. 39-41.
- Madnani, N. y Dorr, B. J. (2010). Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3), pp. 341-387.
- Morante, R. y Daelemans, W. (2009). Learning the scope of hedge cues in biomedical texts. *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, Boulder, Colorado.
- Negri, M. y Kouylekov, M. (2010). A WordNet-based system for multi-way classification of semantic relations. *Proceedings of the 5th International Workshop on Semantic Evaluation*, Los Angeles, California.
- Ó Saghdha, D. y Copestate, A. (2009). Using lexical and relational similarity to classify semantic relations. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Atenas, Grecia.
- Padró, L., Collado, M., Reese, S., Lloberes, M. y Castellón, I. (2010). FreeLing 2.1: Five Years of Open-source Language Processing Tools. *Proceedings of 7th Language Resources and Evaluation Conference*, La Valletta, Malta. Extraído desde <http://www.lsi.upc.edu/~nlp/papers/padro10b.pdf>
- Pang, B., Lee, L. y Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 10.

- Patwardhan, S. (2003). *Incorporating dictionary and corpus information into a context vector measure of semantic relatedness*. Tesis de Maestría, Universidad de Minnesota, Minnesota, Estados Unidos.
- Pedersen, T., Patwardhan, S. y Michelizzi, J. (2004). WordNet::Similarity: measuring the relatedness of concepts. *Proceedings to the Demonstration Papers at HLT-NAAACL 2004*, Boston, Massachusetts, pp. 38-41.
- Real Academia Española. (2001a). Ambiguo. En *Diccionario de la lengua española* (22.<sup>a</sup> ed.). Extraído desde <http://lema.rae.es/drae/?val=ambiguo>
- Real Academia Española. (2001b). Paráfrasis. En *Diccionario de la lengua española* (22.<sup>a</sup> ed.). Extraído desde <http://lema.rae.es/drae/?val=par%C3%A1frasis>
- Real Academia Española (2001c). Racimo. En Diccionario de la lengua española (22.<sup>a</sup> ed.). Extraído desde <http://lema.rae.es/drae/?val=racimo>
- Rennie, J. y Rifkin, R. (2001). *Improving multiclass text classification with the Support Vector Machine*. Massachusetts Institute of Technology, AI Memo AIM.
- Rink, B. y Harabgu, S. (2010). UTD: Classifying semantic relations by combining lexical and semantic resources. *Proceedings of the 5<sup>th</sup> International Workshop on Semantic Evaluation*, Los Angeles, California.
- Salton, G., Wong, A. y Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), pp. 613-620.
- Seung, H. S. y Lee, D. D. (2000). The manifold ways of perception. *Science*, 290(5500), pp. 2268-2269.
- Tsang, V. y Stevenson, S. (2010). A graph-theoretic framework for semantic distance. *Computational Linguistics*, 36(1), pp. 31-69.
- Tejada Camargo, J. (2009). *Construcción automática de un modelo de espacio de palabras mediante relaciones sintagmáticas y paradigmáticas*. PhD Thesis, Instituto Politécnico Nacional, México, D.F.
- Wu, Z. y Palmer, M. (1994). Verb semantics and lexical selection. *Proceedings of the 32<sup>nd</sup> Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, USA, pp. 133-138.
- Xia, R., Zong, C. y Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, 181(6), pp. 1138-1152.
- Yu, Z., Su, L., Li, L., Zhao, Q., Mao, C. y Guo, J. (2010). Question classification based on co-training style semi-supervised learning. *Pattern Recognition Letters*, 31(13), pp. 1975-1980.
- Zhuge, H. (2010). Interactive semantics. *Artificial Intelligence*, 174(2), pp. 190-204.