

# Resolución de anáfora directa basada en conocimiento para pronombres definitivos

Jesus Alexander Alvarado Gutiérrez, Alexander Gelbukh, Olga Kolesnikova

Instituto Politécnico Nacional,  
Centro de Investigación en Computación,  
México

j.alexander.ag@hotmail.com, gelbukh@gelbukh.com,  
kolesolga@gmail.com

**Resumen.** El fenómeno de anáfora es utilizado frecuentemente en la mayoría de los idiomas debido a que permite economizar palabras, permitiendo realizar omisión de antecedentes previamente mencionados en una oración y sustituirlos con referentes, esta investigación se enfoca en la resolución de la relación anafórica presente en los pronombres definitivos especialmente con aquellos que no presentan restricciones lingüísticas de género y cantidad. Actualmente la tendencia inclina a la mayoría de los enfoques a abordar los problemas de procesamiento de lenguaje natural utilizando word embedding por los resultados obtenidos, debido a esto ha perdido popularidad en la utilización de enfoques basados en conocimiento, por lo que en esta investigación se mostrarán resultados de diversos experimentos utilizando este enfoque, mostrando que se pueden obtener resultados representativos. Se proponen diferentes métodos para encontrar los antecedentes correctos en una oración que presenta relación anafórica utilizando las similitudes léxicas basadas en WordNet: calculando las distancias que existen entre cada candidato y el contexto presente en la oración; utilizando la medida de similitud de coseno y añadiendo peso a los candidatos que presenten una frecuencia mayor en las colocaciones con los verbos presentes en la oración.

**Palabras clave.** Pronombres definidos, esquemas de Winograd, resolución de anáfora, wordnet, similitud de coseno.

## Knowledge-based Direct Anaphora Resolution for Definitive Pronouns

**Abstract.** The anaphora phenomenon is the presupposition of an element, so this element is

not mentioned directly, anaphora is a reference to previous antecedent. anaphora resolution tries to find the antecedent of a referent, this research is focused in the anaphoric relation between a noun and a pronoun in the Winograd Scheme, the anaphora resolution is essential for many tasks in Natural Language Processing, such automatic translation, automatic summarization, etc. Nowadays there are many different approaches for anaphora resolution and pronoun resolution, this research is based in a knowledge approach with similarity and relatedness measure. proposing many methods to choose the correct antecedent for a pronoun in anaphoric relation, using stemming, removing determinants, synonyms and hypernyms to get a higher range of match in verb-noun relation.

**Keywords.** Definite pronouns, Winograd schemes, anaphora resolution, cosine similarity.

## 1. Introducción

El problema de resolución de anáfora presente en los pronombres definitivos aun sigue siendo un problema difícil de tratar en la actualidad, debido a la falta de recursos y técnicas que logren realizar una distinción adecuada entre candidatos, esencialmente en el fenómeno de anáfora presente en los esquemas de Winograd [1]:

- $S_1$  : The **bee** landed on the **flower** because it had pollen.
- $P_1$  : it,
- $C_1$  : the bee,

- $C_2$  : the flower,
- $A_1$  : the flower,

donde:

- $S_1$  : Oración (dos antecedentes y un pronombre),
- $P_1$  : Pronombre a desambiguar dentro de la oración  $S_1$ ,
- $C_1$  : Primer antecedente mencionado en la oración  $S_1$ ,
- $C_2$  : Segundo antecedente mencionado en la oración  $S_2$ ,
- $A_1$  : Antecedente correcto para el pronombre  $P_1$ .

Los esquemas de Winograd están basados en pronombres complejos debido a que la oración contiene un **conector de discurso** que la divide en dos partes; en la primer parte se encuentran los dos candidatos y en la segunda se encuentra la referencia para un candidato. El problema recae en que los esquemas de Winograd son propuestos en dos *oraciones pares* [2] las cuales cumplen la misma estructura léxica:

- The bee landed on the flower because it **had** pollen.
- The bee landed on the flower because it **want** pollen.

El problema de resolución de anáfora en pronombres definidos fue propuesto como una alternativa a la prueba de Turing [3] debido a que la prueba de Turing depende de un evaluador humano el cual determinará si sistema esta *pensando* mediante un conjunto de preguntas, a diferencia de los esquemas de Winograd, los cuales no necesitan de un evaluador, ya que estos tienen definido cual es la respuesta correcta lo cual lo vuelve un problema de clasificación binario.

En el trabajo de Altaf Rahman and Vincent Ng [2] se creó un vector de características utilizando diferentes métodos como: Narrative chains, frecuencia de las colocaciones en Google, FrameNet, Heurísticas de polaridad, para crear

un vector de características el cual fue entrenado por una MSV, logrando un 73.05% de exactitud, este experimento conlleva un tiempo relativamente grande, para la extracción de características y finalmente el entrenamiento.

En el presente trabajo se utilizará un vector de características creado a partir de las diferentes medidas de similitud que se pueden extraer de WordNet [4]. aplicadas a las derivaciones semánticas de los sustantivos presentes en el contexto de los candidatos a desambiguar.

## 2. Extracción de derivaciones semánticas

Se realizó un análisis sintáctico de la oración con el analizador Stanford parser[5] para obtener los sustantivos *NN* y verbos *VB* presentes en la oración. Ejemplo:

input:

$S_i(text) =$  the **bee** landed in the **flower**  
because it want **pollen**.

(bee/NN),  
(landed/VBD),  
(flower/NN),  
(want/VBP),  
(pollen/NN)

output:

Se obtuvieron derivaciones semánticas como sinónimos e hiperónimos de cada uno de los sustantivos presentes en la oración. Ejemplo:

input:

$S_i(text) =$  the **bee** landed in the **flower**  
because it want **pollen**.

(bee/NN),  
(animal/NN),  
(insect/NN),  
(landed/VBD),  
(flower/NN),  
(flora/NN),  
(want/VBP),  
(pollen/NN),  
(seed-plant/NN)

output:

### 3. Extracción de similitud semántica

Se aplicaron seis medidas de similitud con todos los sustantivos presentes en el contexto de la oración  $S_i$  incluyendo las derivaciones semánticas (sinónimos e hiperónimos) Con ellos se obtuvo un vector de características de medidas semánticas para cada uno de los candidatos, para ello es necesario definir tres conceptos importantes utilizados en las medidas de similitud.

**Probabilidad de un concepto:** Se calcula con el número de veces que el concepto "c" aparece en el corpus con respecto a la cantidad total de palabras en él (1):

$$P(c) = \frac{\sum_{w \in \text{words}(c)} \text{count}(c)}{N}. \quad (1)$$

**Information Content (IC):** Se refiere a la cantidad de información que tiene un concepto, es calculada con el logaritmo negativo de la probabilidad del concepto como muestra la siguiente fórmula (2):

$$IC(c) = -\log P(c). \quad (2)$$

**Lowest Common Subsumer (LCS):** Es el ancestro común más cercano a dos conceptos tomando en cuenta la jerarquía de WordNet.

El vector de características está formado con siguientes medidas de similitud semántica:

- Wu and Palmer [6]:

$$wup = 2 \times \text{depth}(lcs) / (\text{depth}(s1) + \text{depth}(s2)). \quad (3)$$

- Jiang and Conrath [7]:

$$\text{dist}_{jcn}(c_1, c_2) = 2 \times \log P(LCS(c_1, c_2)) - (\log P(c_1) + \log P(c_2)). \quad (4)$$

- Leacock and Chodorow [8]:

$$lch = -\log(\text{length} / (2 \times D)). \quad (5)$$

- Lin [10]:

$$\text{sim}_{lin}(c_1, c_2) = \frac{2 \times \log P(LCS(c_1, c_2))}{\log P(c_1) + \log P(c_2)}. \quad (6)$$

- Resnik [9]:

$$\text{sim}_{resnik}(c_1, c_2) = IC(LCS(c_1, c_2)) = -\log(LCS(c_1, c_2)). \quad (7)$$

- Path length [11]: Se basa en la intuición de que dos conceptos que pertenecen a la misma jerarquía son similares entre más cercanos estén, esta medida es obtenida contando las aristas entre ellos, tomando en cuenta el camino más corto (Fig. 1).

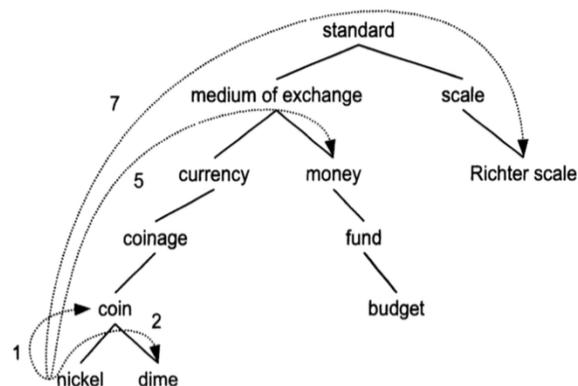


Fig. 1. Ejemplo de la medida de similitud Path length

Estas medidas fueron seleccionadas debido a que lograron mejor desempeño en los experimentos realizados.

### 4. Extracción de concordancia Sustantivo-Verbo

Se obtuvieron dos medidas a partir de la frecuencia de concordancia entre el verbo dentro de la oración con cada uno de los candidatos y sus derivaciones semánticas, para agregarlo al vector de características, las colocaciones de concordancia fueron obtenidas a partir del corpus de Sketch Engine. Una medida fue la frecuencia en la que estas dos palabras aparecen juntas y la segunda medida fue calculada con *tf-idf* (tabla 1).

Tabla 1. Ejemplo de tuplas  $(V, N, frecc, tf, idf, w)$ 

Verbo	Sustantivo	frecuencia	tf	idf	tf-idf
take	place	185576	0.11822153	2.163086	0.25572333
be	part	164582	0.048840113	1.7947271	0.08765467
be	way	133611	0.039649393	1.6282218	0.06455801
be	time	127544	1.7805147E-6	1.4735535	2.6236837E-6
be	thing	83126	0.024667844	1.7624434	0.04347568
take	part	81055	0.051636234	1.7947271	0.092672944
be	member	77505	0.022999799	2.0418725	0.046962656
be	something	77092	0.02287724	2.2324471	0.05107223
be	case	74254	0.022035057	2.1579447	0.047550432
be	year	69744	1.4837623E-6	1.5949645	2.3665482E-6
be	problem	68683	0.02038185	2.0705914	0.042202484
be	place	62293	0.0184856	2.163086	0.03998594
be	number	61279	0.018184694	1.7596849	0.03199933
be	lot	59444	0.017640153	2.2357676	0.039439283
provide	information	58785	0.03744909	2.0086408	0.07522177
provide	service	58505	0.037270717	1.7982997	0.06702392
be	bit	58102	0.017241912	2.079086	0.035847418
be	nothing	51171	0.0151851205	2.8770673	0.043688614
have	time	51145	5.7334664E-6	1.4735535	8.44857E-6
make	decision	50900	0.032425936	2.5711317	0.083371356
be	people	50826	1.7805147E-6	1.4104064	2.5112492E-6
have	effect	50770	0.03234312	2.0464492	0.06618855
have	problem	49299	5.0964145E-6	2.0705914	1.0552592E-5
be	man	48795	0.014480037	1.625207	0.023533056
play	role	47274	0.030115988	2.337194	0.07038691

## 5. Experimentos y resultados

Para los experimentos se utilizó el corpus creado en el trabajo de Altaf Rahman and Vincent Ng [2]: que consta de 941 oraciones pares etiquetadas. Aplicando extracción de medidas de similitud y la extracción de concordancia verbo-sustantivo se crearon vectores de características de cada una de las oraciones dentro del corpus.

Ejemplo de vector de características por medida de similitud (Tabla 2):

—  $S_1$ : the **bee** landed in the **flower** because it want **pollen**,

—  $s$ : frecuencia de concordancia verbo-sustantivo de cada sustantivo,

—  $d$ : frecuencia de concordancia verbo-sustantivo de cada sustantivo y sus sinónimos,

—  $h$ : frecuencia de concordancia verbo-sustantivo de cada sustantivo y sus hiperónimos:

- 1 : medida de similitud  $wup$ ,
- 2 : medida de similitud  $jco$ ,
- 3 : medida de similitud  $lch$ ,
- 4 : medida de similitud  $lin$ ,
- 5 : medida de similitud  $res$ ,
- 6 : medida de similitud  $path$ ,
- $w$  : medida  $tf-idf$ .

Se realizaron tres experimentos con los vectores de características resultantes:

**Tabla 2.** Ejemplo de arreglo de vector de características para cada candidato de la oración  $S_1$ 

vector	s	d	h	1	2	3	4	5	6	w
Candidato 1	0.6000	0.0690	0.4403	0.1936	0.8230	0.1111	0.3160	0.3160	0.4352	0.0002
Candidato 2	0.6315	0.0683	0.5581	0.1920	0.8230	0.1250	0.8960	0.8960	0.2412	0.0005

**Tabla 3.** Ejemplo de arreglo de vectores sometidos a similitud de coseno

vector	s	d	h	1	2	3	4	5	6	w
Candidato 1	0.6000	0.0690	0.4403	0.1936	0.8230	0.1111	0.3160	0.3160	0.4352	0.0002
Ideal	1	1	1	1	1	1	1	1	1	1
Candidato 2	0.6315	0.0683	0.5581	0.1920	0.8230	0.1250	0.8960	0.8960	0.2412	0.0005

- Algoritmo de votación.
- Similitud de coseno.
- Clasificación con una red neuronal.

### 5.1. Algoritmo de votación

El primer experimento que se realizó se obtuvo con la intuición de que el candidato con la medida de similitud menor y la frecuencia de concordancia mayor es el candidato que resuelve la referencia, por ello se implementó el siguiente algoritmo en el cual obtiene el menor valor para cada una de las medidas de similitud y el mayor valor para cada frecuencia de concordancia realizando un conteo con cada una de las medidas, obteniendo como respuesta el candidato con mayor número de votaciones (Fig. 2).

### 5.2. Similitud de coseno

El segundo método se basó en la distancia de coseno en la cual se creó un vector de características ideal, el cual está compuesto por los valores para cada medida en la cual representa una similitud de 1, donde 1 representa el valor ideal y por el lado contrario 0 que no tienen similitud.

Se obtuvieron dos vectores de características para cada candidato y se normalizaron los datos a valores entre 0 y 1.

La intuición de similitud de coseno mide el ángulo entre vectores no la magnitud, por lo que se obtuvo aquel vector de características que maximizaba la similitud de coseno entre los dos vectores de características de cada candidato y el vector ideal (Tabla 3).

**Tabla 4.** Accuracy para cada medida

Medida	Accuracy
w	49.78
s	49.65
2-jco	49.65
3-lch	49.43
h	49.36
d	49.32
6-path	48.96
1-wup	47.80
5-res	45.81
4-lin	44.89

### 5.3. Clasificación con una red neuronal

Para este experimento se creó un vector de características como entrada para la red neuronal que consta de la concatenación de los 2 vectores de características por cada candidato, con un vector resultante de 20 características, para este experimento se tomó el 70% del corpus como conjunto de entrenamiento y el 30% como conjunto de prueba, el conjunto de entrenamiento se sometió a una red neuronal con una capa oculta con 25 nodos, debido a la naturaleza binaria del problema se utilizó una función de activación *relu* en la capa oculta y la función de activación *sigmoid* en la capa de salida (Fig. 3).

## 6. Resultados

Con cada uno de los tres métodos propuestos se realizaron diferentes experimentos, se utilizaron los resultados de cada medida por separado y con

**Algorithm 4.5:** Measure algorithm

```

1  $A \leftarrow null$ ,  $L_n \leftarrow listOfNouns$ ,  $C_1 = S_i(C_1)$ ,  $C_2 = S_i(C_2)$ ,  $Answer = S_i(ans)$ 
2 for each  $noun \subseteq L_n$  then:
3   if  $getMeasures(noun, C_1) > getMeasures(noun, C_2)$  then:
4      $A \leftarrow C_1$ 
5   end if
6   else  $getMeasures(noun, C_2) > getMeasures(noun, C_1)$  then:
7      $A \leftarrow C_2$ 
8   end else
9 end for
10 if  $Answer == A$  then:
11   return 1
12 end if
13 else:
14   return 0
15 end else

```

Fig. 2. Algoritmo de votación que aumenta la votación en 1 para cada medida que cumpla la condición

Tabla 5. Precision para cada medida

Medida	Precision
w	50.41
d	50.00
h	50.00
s	49.73
2-jco	49.73
3-lch	49.73
6-path	48.64
1-wup	48.01
5-res	45.45
4-lin	42.00

Tabla 6. Recall para cada medida

Medida	Recall
h	52.08
4-lin	51.12
w	50.83
s	49.77
d	48.44
2-jco	43.37
3-lch	42.53
1-wup	42.17
6-path	41.66
5-res	40.00

las posibles combinaciones, por lo que se realizó el conjunto potencia de las 10 medidas para realizar los experimentos realizando un total de  $10^2 - 1$  experimentos, para encontrar el mejor resultado cada experimento se evaluó con las siguientes medidas utilizadas en clasificación: precisión (eq: 8), recall (eq: 9) y F1:

$$precision = \frac{TP}{TP + FP}, \quad (8)$$

donde:

- $TP$  (True Positives): representa los datos positivos clasificados como positivos.
- $FP$  (False Positives): representa los datos clasificados como positivos que son falsos.

Recall es dada por (eq:9):

$$recall = \frac{TP}{TP + FN}, \quad (9)$$

donde:

**Tabla 7.** F1 para cada medida

Medida	F1
h	51.02
w	50.62
s	49.77
d	49.20
2-jco	46.34
3-lch	45.85
1-wup	44.90
6-path	44.88
5-res	42.55
4-lin	40.00

**Tabla 8.** F1 para la combinación de las medidas de similitud

Medida	F1
[s, 3, d, 6, h]	51.36
[w, s, 3, d, 6]	51.36
[1, w, s, 3, d, 6, h]	51.52
[1, w, s, 5, h]	51.46
[1, w, s, 6, h]	51.42

- *TP* (True Positives): representa los datos positivos clasificados como positivos.
- *FN* (False Negative): representa los datos falsos clasificados como positivos.

### 6.1. Resultados por combinación

Con los valores de cada una de las medidas se formo un vector de características por cada candidato los cuales fueron sometidos a los 3 experimentos previamente mencionados: algoritmo de votación con combinación (tabla 8), similitud de coseno con combinación (tabla 9), y finalmente el resultado del entrenamiento con la red neuronal por combinación de medidas (tabla 10), con los que se obtuvo los siguientes resultados (se muestran los resultados mas representativos por combinación), se realizó la evaluación con la métrica F1 debió a que otros métodos del estado del arte han sido evaluados con esta métrica.

Finalmente se realizó una comparativa entre los trabajos que han sido probados con el mismo

**Tabla 9.** F1 para la similitud de coseno

Medida	F1
[1, 5, 6, s, d, w]	55.65
[1, 6, h, s, r]	54.34
[1, w, 3, d, 6, h]	54.32
[1, w, s, ]	54.46
[1, w, s, 6, h]	54.82

**Tabla 10.** F1 red neuronal

Key	F1
[1-2-5-6-s-d-h]	61.53
[1-3-6]	61.53
[1, w, 3, d, 6, h]	60.89
[1-3-4-5-6-s-d]	60.89
[1-2-3-5-h]	60.89

corpus (Tabla 11), obteniendo mejor resultado la mayoría de los métodos simples utilizados en el estado del arte y muy cercanos a los resultados obtenidos por trabajos que utilizan modelado con redes neuronales para extracción de características, entrenamiento y clasificación, por lo que nuestro sistema es eficiente en tiempo y ejecución.

### 6.2. Resultados por medida

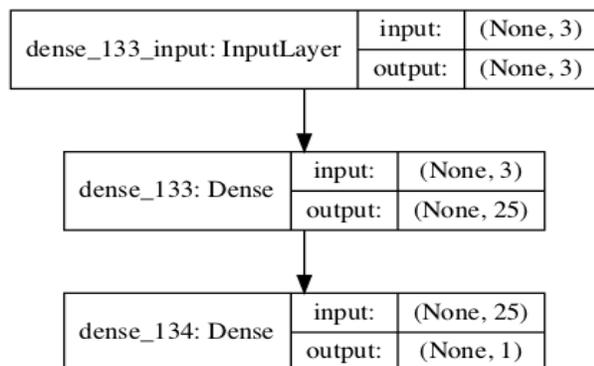
Se muestran los resultados individuales por cada una de las 6 medida de similitud y las 3 frecuencias de concordancia: accuracy (tabla 4), precision (tabla 5), recall (tabla 6) y F1 (tabla 7), ordenados por la medida que obtuvo mejor desempeño individualmente.

## 7. Conclusiones

Los resultados obtenidos en los experimentos como resultado de las tres nuevas técnicas basadas en conocimiento, muestran que es posible obtener resultados competitivos, creando un vector de características basada en conocimiento evitando utilizar demasiados recursos y tiempo de ejecución como otros metodos del estado del arte que para la extracción

**Tabla 11.** Valores de exactitud en Winograd schema challenge

Autor	Name	F1
(Rahman and Ng et al., 2012)	Narrative Chains	30.67
(Rahman and Ng et al., 2012)	Google	33.16
(Lee et al., 2011)	The Stanford resolver.	40.1
(quian Liu at al., 2016)	USSM	48.7
(quian Liu at al., 2016)	NKAM	49.1
-	The Random baseline.	50
(quian Liu at al., 2016)	USSM + NKAM	50.2
(Durrett and Klein, 2013)	BERKELEYnew	50.32
(Chang et al.2013),	Illinois	51.48
-	<b>our system - similitud de características</b>	<b>51.52</b>
-	<b>our system - similitud de coseno</b>	<b>55.65</b>
-	<b>our system - red neuronal</b>	<b>61.53</b>
(Canasai Kruengkrai, 2014)	-	69.68
(Rahman and Ng et al., 2012)	MENTRANKER	73.05
(Peng and Khashabi, 2015 )	KnowComb	76.76

**Fig. 3.** Modelo utilizado para entrenar la red neuronal

características utilizan redes neuronales o word embedding.

Los resultados son alentadores, indicando que es posible agregar mas restricciones léxicas y semánticas, para evaluar diferentes condiciones que pueden presentar las oraciones como por ejemplo módulos para tratar con negación, con marcos de roles, antónimos, merónimos, holónimos, etc. que brindarán mas información semántica a los vectores de características.

Por lo que para trabajos futuros se planea agregar estas características y adaptar este

vector de características para utilizarlo con word embedding.

## Referencias

1. **Levesque, H.J. (2011).** The Winograd Schema Challenge. Proceedings of the Twenty-Seventh Conference on Innovative Applications of Artificial Intelligence, pp. 63–68
2. **Rahman, A., Ng, V. (2012).** Resolving complex cases of definite pronouns: the Winograd schema challenge. Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 777–789, Association for Computational Linguistics
3. **Turing, A.M. (1950).** Computing machinery and intelligence. *Mind*, 59(236), pp. 433–460.
4. **Fellbaum, C. (1998).** WordNet: An electronic database.
5. **De Marneffe, M.C., Manning, C.D. (2008).** Stanford typed dependencies manual. pp. 338–345, Technical report, Stanford University.
6. **Wu, Z., Palmer, M. (1994).** Verbs semantics and lexical selection. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pp. 133–138, Association for Computational Linguistics.

7. **Jiang, J.J., Conrath, D.W. (1997).** Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008.
8. **Leacock, C., Chodorow, M. (1998).** Combining local context and WordNet similarity for word sense identification. In: WordNet: An electronic lexical database, pp. 265–283.
9. **Resnik, P. (1995).** Using information content to evaluate semantic similarity in a taxonomy. arXiv preprint cmp-lg/9511007.
10. **Lin, D. (1997).** Using syntactic dependency as local context to resolve word sense ambiguity. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, pp. 64–71, Association for Computational Linguistics.
11. **Pedersen, T., Patwardhan, S., Michelizzi, J. (2004).** WordNet::Similarity: measuring the relatedness of concepts. Demonstration papers at HLT-NAACL 2004, pp. 38–41, Association for Computational Linguistics

*Article received on 10/05/2020; accepted on 11/12/2020.  
Corresponding author is Alexander Gelbukh.*