Algunas semánticas lógicas para reconocer implicación textual

José-de-Jesús Lavalle-Martínez[1,2 *], Manuel Montes-y-Gómez[1], Héctor Jiménez-Salazar[3], Luis Villaseñor-Pineda[1], David Pinto-Avendaño[2]

[1] Coordinación de Ciencias Computacionales
Instituto Nacional de Astrofísica, óptica y Electrónica
[2] Facultad de Ciencias de la Computación
Benemérita Universidad Autónoma de Puebla
[3] Departamento de Tecnologías de la Información
Universidad Autónoma Metropolitana-Cuajimalpa

Resumen En el presente trabajo se presentan algunos métodos lógicos que en general se han utilizado para el procesamiento de lenguaje natural y últimamente en la tarea para reconocer implicación textual, estos son: Teoría de Representación de Discurso, Gramáticas Categoriales, Lógica de Predicados Dinámica y Lógica Natural. El interés principal es reunir, en un solo artículo en español, los principales formalismos lógicos que se han utilizado para atender aspectos semánticos propios del procesamiento de lenguaje natural.

Palabras clave: Teoría de Representación de Discurso, Gramáticas Categoriales, Lógica de Predicados Dinámica, Lógica Natural, Semántica de Lenguaje Natural.

1. Introducción

Si bien en el *Procesamiento de Lenguaje Natural* (NLP) el enfoque lógico no ha sido preferido, debido principalmente al mejor desempeño que han mostrado otros enfoques [1], en la tarea de *Implicación Textual* se ha vuelto a retomar [2]. En primer lugar porque la lógica es quien formaliza la noción de implicación; en segundo lugar porque, gracias a la especificación y verificación formal de software y hardware, se han desarrollado herramientas [3] tanto teóricas como prácticas que han permitido razonar sobre sistemas de tamaño industrial; en tercer lugar dado que en la tarea de implicación textual ningún enfoque ha demostrado supremacía sobre los demás [4,5].

De acuerdo con el Portal de Implicación Textual de la ACL, la *Implicación Textual* es una relación direccional entre dos fragmentos de texto. La relación se cumple siempre que la veracidad del segundo fragmento de texto se sigue de la veracidad del primer fragmento de texto. En el marco de la Implicación Textual

[*] Este trabajo ha sido apoyado por PROMEP mediante el convenio PROMEP/103.5/13/5618 y número de becario BUAP-803.

al texto implicante se le llama *texto* (t) y al texto implicado se le llama *hipótesis* (h).

Dos son los principales enfoques lógicos que se han usado para el reconocimiento de implicación textual.

Por un lado está el de Bos y Markert [6] quienes utilizan Gramáticas Categoriales (CG), Teoría de Representación de Discurso (DRT), Demostradores Automáticos de Teoremas y Chequeadores de Modelos para reconocer cuando hay implicación textual entre el par (t, h).

Por otro lado está el trabajo de MacCartney y Manning [7], ellos utilizan Lógica Natural para el reconocimiento de la implicación textual.

Ambos métodos tienen ventajas y desventajas: Bos y Markert logran buena precisión pero mala cobertura; MacCartney y Manning, por el contrario, logran buena cobertura pero mala precisión.

Desde un punto de vista más técnico los diversos métodos usados tienen también ventajas y desventajas: la DRT tiene como ventaja que fue creada para solucionar los problemas de anáfora que las gramáticas de Montague no podían manejar [8], pero tiene el inconveniente de que su salida es una fórmula de la lógica de primer orden, por tanto razonar sobre su validez es indecidible en general.

Las CG [9] tienen la ventaja de que sintaxis y semántica están íntimamente ligadas y que la semántica se encuentra evaluando una expresión del cálculo lambda, su principal desventaja es que no pueden manejar el fenómeno de la anáfora.

La Lógica de Predicados Dinámica (DPL) [10] fue creada inicialmente para razonar sobre los cambios de estado que tiene un programa imperativo, posteriormente se ha utilizado para tratar la anáfora.

Por último, la Lógica Natural [11] tiene la desventaja de que no puede tratar con el fenómeno de la anáfora creemos que por ello es baja en precisión, su ventaja es que es una teoría decidible.

En este trabajo se presentan algunas teorías lógicas que se han utilizado en el procesamiento del lenguaje natural en general y, particularmente, en el reconocimiento de implicación textual.

2. Teoría de Representación de Discurso

La Teoría de Representación de Discurso [12,8], es una de las teorías para semántica dinámica. El interés principal de estas teorías es tomar en cuenta la dependencia del contexto que tiene el significado. Es una característica ubicua de los lenguajes naturales que lo que se expresa es interpretable sólo cuando el intérprete toma en cuenta el contexto en el que se hizo, el significado de lo expresado depende del contexto.

Aún más, la interacción entre el contexto y lo expresado es recíproco. Cada expresión contribuye (vía la interpretación que se le da) al contexto en el que se hace. Modifica el contexto en un nuevo contexto, en el que esta contribución se refleja, es este nuevo contexto el que influirá en la interpretación de cualqueir cosa que se exprese posteriormente.

2.1. Definición

Una Estructura de Representación de Discurso [13] se obtiene a partir de un conjunto de constantes de relación, un conjunto de constantes individuales y un conjunto infinito de variables individuales (a las constantes y variables individuales se les llama *referentes de discurso*). Las condiciones y las cajas se construyen a partir de estos conjuntos y de las siguientes cláusulas:

- Si R es una relación constante de aridad n y $\delta_1, \ldots, \delta_n$ son referentes de discurso entonces $R(\delta_1, \ldots, \delta_n)$ es una condición;
- Si δ_1 y δ_2 son referentes de discurso entonces δ_1 **is** δ_2 es una condición.
- Si K_1 y K_2 son cajas, entonces **not** K_1, K_1 **or** K_2 y $K_1 \Rightarrow K_2$ son condiciones;
- Si $\gamma_1, \ldots, \gamma_m$ son condiciones ($m \geq 0$) y $x_1, \ldots, x_n$ son variables ($n \geq 0$) entonces $[x_1 \ldots x_n | \gamma_1, \ldots, \gamma_m]$ es una caja.
- Si K_1 y K_2 son cajas entonces $K_1; K_2$ es una caja.

El lenguaje generado por las cláusulas anteriores se interpreta como modelos ordinarios de primer orden. Los modelos se definen como pares $< D, I >$, donde D es un conjunto arbitrario no vacío e I es una función que tiene como dominio el conjunto de constantes tal que $I(c) \in D$ para cada constante individual c e $I(R) \subseteq D^n$ para cada constante de relación R de aridad n.

Una *asignación* para tal modelo de primer orden $M =< D, I >$ es una función del conjunto de variables de referentes de discurso al dominio D. Escribimos $a[x_1, \ldots, x_n]a'$ como una abreviación de "las asignaciones a y a' difieren a lo más en sus valores para $x_1, \ldots, x_n$. Como es usual, definimos $\|\delta\|^{M,a}$ como $a(\delta)$ si δ es una variable y como $I(\delta)$ si δ es una constante.

Las cláusulas siguientes definen el valor semántico $\|\gamma\|^M$ de una condición γ en un modelo M como un conjunto de asignaciones, el valor semántico $\|K\|^M$ de una caja K en M se define como una relación binaria entre asignaciones (el super índice M se omitirá).

- $\|R(\delta_1, \ldots, \delta_n)\| = \{a \mid \langle \|\delta_1\|^a, \ldots, \|\delta_n\|^a \rangle \in I(R)\}$
- $\|\delta_1 \textbf{ is } \delta_2\| = \{a \mid \|\delta_1\|^a = \|\delta_2\|^a\}$
- $\|\textbf{not } K\| = \{a \mid \neg \exists a' \langle a, a' \rangle \in \|K\|\}$
- $\|K_1 \textbf{ or } K_2\| = \{a \mid \exists a'(\langle a, a' \rangle \in K_1 \vee \langle a, a' \rangle \in K_2)\}$
- $\|K_1 \Rightarrow K_2\| = \{a \mid \forall a'(\langle a, a' \rangle \in K_1 \rightarrow \exists a'' \langle a', a'' \rangle \in K_2)\}$
- $\|[x_1 \ldots x_n | \gamma_1, \ldots, \gamma_m]\| = \{\langle a, a' \rangle \mid a[x_1, \ldots, x_n]a' \wedge a' \in \|\gamma_1\| \cap \ldots \cap \|\gamma_m\|\}$
- $\|K_1; K_2\| = \{\langle a, a' \rangle \mid \exists a''(\langle a, a'' \rangle \in \|K_1\| \wedge \langle a'', a' \rangle \in \|K_2\|)\}$

Una caja K es *verdadera* en un modelo M bajo una asignación a si y sólo si existe alguna asignación a' tal que $\langle a, a' \rangle \in \|K\|^M$; una condición γ es verdadera en M bajo a si y sólo si $a \in \|\gamma\|^M$.

2.2. Ejemplo

Dado el siguiente texto en inglés "*A man adores a woman. She abhors him.*" La caja cerrada que le corresponde a la sentencia "*A man adores a woman.*" es (1), la única caja razonable que se puede asociar con la sentencia abierta "*She abhors him.*" es la caja abierta (2) la cual es verdadera bajo una asignación a si y sólo si la condición x_2 *abhors* x_1 es verdadera bajo a.

Los pronombres anafóricos "*she*" y "*him*" obtienen cualquier valor que la asignación de entrada asocie con los referentes de discurso x_2 y x_1. La caja (2) se puede interpretar como una prueba: dada cualquier asignación de entrada a, prueba si $a(x_2)$ *abhors* $a(x_1)$, si es así regresa a a como salida, si no la prueba falla y ninguna salida se regresa.

$$[x_1\ x_2 | man\ x_1, woman\ x_2, x_1\ adores\ x_2] \tag{1}$$

$$[| x_2\ abhors\ x_1] \tag{2}$$

De tal manera que la caja correspondiente al texto "*A man adores a woman. She abhors him.*" es:

$$[x_1\ x_2 | man\ x_1, woman\ x_2, x_1\ adores\ x_2]; [| x_2\ abhors\ x_1]$$

3. Gramáticas Categoriales

Las gramáticas categoriales [9] son una forma de gramáticas lexicalizadas, en las que la aplicación de las reglas sintácticas está condicionada completamente por el tipo sintáctico, o la *categoría* de sus entradas.

Las categorías identifican sus constituyentes como *categorías primitivas* o *funciones*. Las categorías primitivas, tales como N, NP, PP, S, etc, pueden enriquecerse con características tales como número, caso, inflección y similares.

Las funciones (tales como los verbos) portan categorías que identifican el tipo de su resultado y el de sus argumentos/complementos (ambos pueden ser a su vez funciones o categorías primitivas). Las categorías función también definen el orden en el que los argumentos se deben combinar y si ocurren a la derecha o la izquierda del funtor.

Cada categoría sintáctica se asocia con una forma lógica cuyo tipo semántico está determinado completamente por la categoría sintáctica.

En gramáticas categoriales la información sintáctica, de la clase que se puede capturar para el Inglés mediante reglas de producción como (3), (4) y (5), se transfiere a entradas léxicas como (6):

$$S \rightarrow NP \quad VP \tag{3}$$

$$VP \rightarrow TV \quad NP \tag{4}$$

$$TV \rightarrow \{proved, finds, \ldots\} \tag{5}$$

$$proved := (S \backslash NP)/NP \tag{6}$$

Esta categoría sintáctica identifica al verbo transitivo como una función, especifica el tipo y direccionalidad de sus argumentos y el tipo de su resultado. Aquí se usa la notación del "resultado más izquierdo" en la que una función que *combina a la derecha* sobre un dominio β en un rango α se escribe α/β, el correspondiente funtor que *combina a la izquierda* se escribe $\alpha\backslash\beta$, donde α y β también pueden ser categorías función.

3.1. Definición del Lenguaje Categorial

En Gramáticas Categoriales los *tipos* (también llamados *categorías*) se definen como sigue:

$$L ::= P|(L/L)|(L\backslash L)$$

donde P es el conjunto de tipos primitivos, los cuales son llamados tipos atómicos o categorías básicas, los tipos más usuales son S (para sentencias), NP (para frases nominales), y puede incluir PP (para frases preposicionales), INF (para infinitivos), etc.

Es usual decir que una fórmula de tipo X/Y o $X\backslash Y$ es un funtor, siendo la fórmula Y su argumento y la fórmula X su resultado.

3.2. Reglas de aplicación funcional, categorías sintácticas

Para permitir que funtores como (6) combinen con sus argumentos necesitamos reglas combinatorias, de éstas las dos más simples son las reglas de aplicación funcional siguientes:

$$\frac{X/Y \quad Y}{X} > \qquad\qquad \frac{Y \quad X\backslash Y}{X} <$$

3.3. Ejemplo de categorías sintácticas

Dadas la oración "*Marcel proved completeness*", y las siguientes entradas léxicas $Marcel$:= NP, *proved* := $(S\backslash NP)/NP$ y *completeness* := NP, se procede como sigue: dado que $Marcel$ y *completeness* son elementos del tipo primitivo NP, no se les puede aplicar alguna de las reglas de aplicación funcional, pero notamos que *proved* tiene el tipo funcional $(S\backslash NP)/NP$ que es el tipo de los verbos transitivos.

Así, para combinar el tipo funcional $(S\backslash NP)/NP$ necesitamos a la derecha de *proved* un elemenot del tipo NP en este caso *completeness*, usando la regla $>$ obtenemos el tipo funcional $(S\backslash NP)$. Para combinar este tipo mediante la regla $<$ necesitamos un elemento de NP a la izquierda de *proved*, en este caso tenemos a $Marcel$, aplicando dicha regla obtenemos el tipo S que es el tipo de las oraciones sintácticamente correctas.

El razonamiento anterior se puede expresar gráficamente mediante un árbol de derivación como el siguiente (al final de las líneas horizontales se marca que regla se usó para la derivación, se marca con Lex cuando se usa una entrada léxica).

$$\dfrac{\dfrac{Marcel}{NP}\,\text{Lex} \qquad \dfrac{\dfrac{proved}{(S\backslash NP)/NP}\,\text{Lex} \quad \dfrac{completeness}{NP}\,\text{Lex}}{S\backslash NP}\,>}{S}\,<$$

3.4. Reglas de Aplicación Funcional, Tipos Semánticos

Se puede considerar que las categorías codifican el tipo semántico de su traducción. La traducción se puede hacer explícita asociando una forma lógica con la categoría sintáctica completa a través del operador :, se asume que éste tiene menor precedencia que los operadores categoriales / y \.

Por supuesto se deben de expandir las reglas de aplicación funcional de acuerdo a los tipos semánticos, recordando que los operadores / y \ definen categorías funcionales el resultado de aplicar las reglas > y <, cuando ya se tienen los tipos semánticos, debe ser la evaluación de una función, como se indica a continuación.

$$\frac{X/Y : f \quad Y : a}{X : fa}\,> \qquad\qquad \frac{Y : a \quad X\backslash Y : f}{X : fa}\,<$$

3.5. Ejemplo de tipos semánticos

Enriqueciendo con tipos semánticos las entradas léxicas del ejemplo "*Marcel proved completeness*" se tiene

$$\begin{aligned} Marcel &:= NP : marcel' \\ proved &:= (S\backslash NP)/NP : \lambda x.\lambda y.prove'xy \\ completeness &:= NP : completeness' \end{aligned}$$

Como se observa las categorías funcionales se enriquecen semánticamente mediante cálculo lambda y las categorías primitivas mediante una constante que se forma primando la entrada léxica, pero sin considerar su tipo sintáctico.

Aplicado la regla > para tipos semánticos a $(S\backslash NP)/NP : \lambda x.\lambda y.prove'xy$ y $NP : completeness'$ obtenemos $S\backslash NP : \lambda y.prove'completeness'y$. Si ahora le aplicamos la regla < a dicho resultado y a $NP : marcel'$, se obtiene $S : prove'completeness'marcel'$. Nuevamente este razonamiento se puede expresar mediante el siguiente árbol de derivación semántica.

$$\dfrac{\dfrac{Marcel}{NP : marcel'}\,\text{Lex} \qquad \dfrac{\dfrac{proved}{(S\backslash NP)/NP : \lambda x.\lambda y.prove'xy}\,\text{Lex} \quad \dfrac{completeness}{NP : completeness'}\,\text{Lex}}{S\backslash NP : \lambda y.prove'completeness'y}\,>}{S : prove'completeness'marcel'}\,<$$

4. Lógica Dinámica de Predicados

En DPL [10] la dinámica se refiere a la información sobre cosas que se van introduciendo en un discurso y que sirven como posibles antecedentes para pronombres anafóricos subsecuentes. Como es usual en lingüística, las frases nominales (frases nominales indefinidas y pronombres) se asocian con índices o variables, para poder indicar casos de correferencia y ligado. La información relevante es información sobre los posibles valores de dichas variables, las cuales pueden ser cambiadas y actualizadas conforme avanza el discurso.

4.1. Definición

Se define $[\![\phi]\!]_M$, la interpretación de una fórmula ϕ de la lógica de predicados de primer orden relativa a un modelo ordinario M de la lógica de predicados de primer orden, como un conjunto de pares de asignaciones a variables, asignaciones entrada/posible-salida $\langle g, h\rangle$. La idea es que un par $\langle g, h\rangle$ está en la interpretación de ϕ relativa a M si y sólo si sobre la asignación de entrada g la fórmula ϕ puede interpretarse exitosamente y da como posible-salida la asignación h. Si no hacen falta se omitirán las referencias a M.

Un lenguaje L para DPL es el de la lógica de predicados de primer orden ordinaria, basado en un conjunto C de constantes individuales c y conjuntos R^n de constantes relacionales R de aridad n y un conjunto numerable de variables V. El conjunto de términos $T = C \cup V$ consiste de las constantes individuales y las variables del lenguaje, las fórmula atómicas $Rt_1 \dots t_n$ se componen de predicados R de aridad n y una secuencia de n términos $t_1, \dots, t_n$, también pueden ser de la forma $t_i = t_j$, enunciando la identidad de los valores de los términos t_i y t_j. Las fórmulas se construyen a partir de las fórmulas atómicas usando negación $(\neg)$, cuantificadores existencial y universal $(\exists x, \forall y)$, conjunción $(\wedge)$, disyunción $(\vee)$ e implicación $(\rightarrow)$.

Un modelo $M = \langle D, V\rangle$ es un modelo usual de primer orden con un dominio de individuos D y una función de interpretación V para las constantes individuales y relacionales de nuestro lenguaje. La función V asigna un individuo $V(c) \in D$ a las constantes individuales de L y un conjunto de $n-$tuplas de individuos $V(R^n) \subseteq D^n$ a sus constantes relacionales de aridad n. En la interpretación de DPL también usamos asignaciones variables f, g, h, k, l las cuales asignan individuos $f(x) \in D$ a las variables $x \in V$, así que son funciones de V a D. La interpretación $[t]_{M,g}$ de un término t en un modelo M y relativo a una asignación g es $V(t)$ si t es una constante individual y $g(t)$ si t es una variable.

Usamos $g[x/d]$ para la asignación variable h que es como g excepto que asigna d a x, así para toda $y \in V$, si $x \neq y$ entonces $g[x/d](y) = g(y)$ y si $x = y$ entonces $g[x/d](y) = d$. Escribimos $g[x]h$ si y sólo si $h = g[x/d]$ para algún individuo d y $g[X]h$ si y sólo si $X = \{x_1, \dots, x_n\}$ y existen $k_1, \dots, k_{n-1}$ tal que $g[x_1]k_1, \dots, k_{n-1}[x_n]h$. Usando estos dispositivos notacionales podemos enunciar la semántica de DPL como sigue:

- $[\![Rt_1 \dots t_n]\!]_M = \{\langle g, h\rangle \mid g = h \wedge \langle [t_1]_{M,g}, \dots, [t_n]_{M,g}\rangle \in V(R)\}$
- $[\![t_i = t_j]\!]_M = \{\langle g, h\rangle \mid g = h \wedge [t_i]_{M,g} = [t_j]_{M,g}\}$
- $[\![\neg\phi]\!]_M = \{\langle g, h\rangle \mid g = h \wedge \text{ para ningún } k : \langle g, k\rangle \in [\![\phi]\!]_M\}$
- $[\![\exists x\phi]\!]_M = \{\langle g, h\rangle \mid \text{ para algún } k : g[x]k \wedge \langle k, h\rangle \in [\![\phi]\!]_M\}$
- $[\![\forall x\phi]\!]_M = \{\langle g, h\rangle \mid g = h \wedge \text{ para todo } k : \text{ si } g[x]k \text{ entonces existe } h : \langle k, h\rangle \in [\![\phi]\!]_M\}$
- $[\![\phi \wedge \psi]\!]_M = \{\langle g, h\rangle \mid \text{ para algún } k : \langle g, k\rangle \in [\![\phi]\!]_M \wedge \langle k, h\rangle\} \in [\![\psi]\!]_M\}$
- $[\![\phi \vee \psi]\!]_M = \{\langle g, h\rangle \mid g = h \wedge \text{ para algún } k : (\langle g, k\rangle \in [\![\phi]\!]_M \vee \langle g, k\rangle \in [\![\psi]\!]_M)\}$
- $[\![\phi \rightarrow \psi]\!]_M = \{\langle g, h\rangle \mid g = h \wedge \text{ para todo } k : \text{ si } \langle g, k\rangle \in [\![\phi]\!]_M \text{ entonces existe } h : \langle k, h\rangle \in [\![\psi]\!]_M\}$

4.2. Ejemplo de Interpretación Dinámica de un Discurso

Considere el siguiente texto en inglés "*A farmer owned a donkey. It was unhappy. It didn't have a tail.*", a éste le corresponde la siguiente fórmula de la lógica dinámica de predicados

$$\exists x(Fx \wedge \exists y(Dy \wedge Oxy)) \wedge (Uy \wedge \neg\exists z(Tz \wedge Hyz))$$

Relativo a la asignación de entrada g se tendrá un asignación de salida h si podemos encontrar asignaciones k y l tales que k es una salida posible al interpretar $\exists x(Fx \wedge \exists y(Dy \wedge Oxy))$ relativo a g, y l es una salida posible al interpretar Uy relativo a k, y h es una salida posible al interpretar $\neg\exists z(Tz \wedge Hyz)$ relativo a l.

Ya que la segunda fórmula es atómica y la tercera una negación, sabemos que en este caso $k = l$ y $l = h$. La asignación k (esto es: h) se obtiene de g al reiniciar el valor de x tal que $k(x) = h(x) \in I(F)$, y después reiniciando el valor de y tal que $k(y) = h(y) \in I(D)$ y $\langle h(x), h(y)\rangle \in I(O)$. Esto es, $h(x)$ es un granjero que posee un burro $h(y)$. Observe que para cualquier granjero f y burro d que f posee, existe una asignación correspondiente $h' : g[\{x, y\}]h'$ y tal que $h(x) = f$ y $h(y) = d$.

El segundo elemento de la conjunción primero prueba si y es infeliz, esto es, si $l(y) = k(y) = h(y) \in I(U)$. El tercer elemento de la conjunción, una negación, prueba si la asignación h no puede servir como entrada para satisfacer la fórmula interna $\exists z(Tz \wedge Hyz)$. Esta subfórmula se satisface relativa a h si y sólo si existe una asignación h' tal que $h[z]h'$ y $h'(z) \in I(T)$ y $\langle h'(y), h'(z)\rangle \in I(H)$, esto es, si y sólo si podemos cambiar la valuación h de z en cualquier cosa que tenga cola mediante $h(y)$.

La negación de la subfórmula prueba si no podemos cambiar la valuación de z en dicha manera. Juntando todo, $\langle g, h\rangle$ está en la interpretación de nuestro ejemplo si y sólo si $g[x, y]h$ y $h(x)$ es un granjero que posee un burro $h(y)$ el cual es infeliz y no tiene cola. Observe, una vez más, que para cualquier granjero f y burro sin cola e infeliz d al que f posee, existe una asignación correspondiente $h' : g[x, y]h'$ tal que $h(x) = f$ y $h(y) = d$.

5. Lógica Natural

Para desarrollar un enfoque cognitivo de razonamiento es prometedor factorizar el aspecto sintáctico (aquel que tiene que ver con coincidencia de patrones y estructuras sintácticas) del resto. Un candidato obvio para esta tarea es el llamado cálculo de monotonía o cálculo de la lógica natural [11]. Este cálculo tiene un lado sintáctico y un lado semántico.

El fundamento semántico del razonamiento monótono es una generalización de la noción de consecuencia lógica para tipos arbitrarios, lo cual se logra definiendo órdenes parciales $\Longrightarrow$ sobre todos los tipos (no sólo el tipo de oraciones, también el de frases verbales, predicados, adjetivos, cuantificadores, etc.).

En estos términos se puede definir que significa que una función del tipo α en el tipo β *preserve el orden* o *invierta el orden*. Las funciones que preservan el orden son las funciones f tales que si $x \Longrightarrow y$ entonces $f(x) \Longrightarrow f(y)$. Las funciones que invierten el orden son las funciones f tales que si $x \Longrightarrow y$ entonces $f(y) \Longrightarrow f(x)$.

El lado sintáctico del cálculo de monotonía tiene que ver con marcar la monotonía de las componentes de una estructura sintáctica. Sea S una estructura sintáctica, y sea A una componente de esa estructura. Suponga que A tiene tipo α y que S tiene tipo β. Considere la función sintáctica F que consiste en reemplazar la componente A por otra componente adecuada de tipo α. En otras palabras, considere la función $F = \lambda Y.S[Y/A]$. Entonces la contraparte semántica de F es una función f de tipo $\alpha \rightarrow \beta$. La validez y completitud de un cálculo de monotonía tiene que ver con la relación entre F y f.

Un algoritmo de marcación de monotonía es *válido* si se cumple lo siguiente: si A se marca + en S entonces la función que interpreta $\lambda Y.S[Y/A]$ preserva la monotonía, si A se marca - en S entonces la función que interpreta $\lambda Y.S[Y/A]$ invierte el orden.

Un algoritmo de marcación de monotonía es *completo* si se cumple lo siguiente: si la función que interpreta $\lambda Y.S[Y/A]$ preserva el orden entonces A se marca + en S, si la función que interpreta $\lambda Y.S[Y/A]$ invierte el orden entonces A se marca - en S.

5.1. Semántica de monotonía

Así como podemos decir que "*Gaia is smiling*" implica lógicamente "*Gaia is smilling or Gaia is crying*", nos gustaría decir que "*smilling*" implica lógicamente "*smilling or crying*", o que "*dancing*" implica lógicamente "*moving*", también que "*at least three*" implica lógicamente "*at least two*", etc.

"*Gaia is smilling*" es una oración, "*smilling*" es un predicado, "*at least three*" es un cuantificador. Sabemos que una sentencia implica a otra si siempre que la primera es verdadera la segunda también lo es. La manera obvia de trasladar esta noción a predicados es estipulando que un predicado implica a otro si se cumple que para todo sujeto la oración que se obtiene al combinar un sujeto con el primer predicado implica a la oración que se obtiene al combinar ese mismo sujeto con el segundo predicado. Similarmente para cuantificadores, para obtener una oración a partir de "*at least three*", se tiene que combinar el cuantificador con un sustantivo y un verbo. Ya que en verdad se cumple que para todo sustantivo N y verbo V "*at least three N V*" implica "*at least two N V*", podemos decir que "*at least three*" implica "*at least two*".

Iniciaremos con los tipos básicos t (valores veritativos, el tipo de las oraciones) y e (entidades, el tipo de los nombres propios). Los tipos complejos se definen por recursión como sigue:

1. e y t son tipos,
2. si α y β son tipos, entonces $\alpha \rightarrow \beta$ es un tipo.

La relación de implicación se define como sigue (usamos $E :: \alpha$ para expresar que "la expresión sintáctica E tiene tipo semántico α"):

1. Si $E, E' :: e$ entonces $I(E) \Longrightarrow I(E')$ si $I(E) = I(E')$,
2. Si $E, E' :: t$ entonces $I(E) \Longrightarrow I(E')$ si $I(E) \leq I(E')$,
3. Si $E, E' :: \alpha \to \beta$ entonces $I(E) \Longrightarrow I(E')$ si y sólo si para todo $x \in D_\alpha, I(E)(x) \Longrightarrow I(E')(x)$.

Aquí $I(E)$ denota la interpretación de E, y D_α se usa para el dominio de objetos de tipo α. Si $E :: \alpha$ entonces $I(E) \in D_\alpha$, es decir, la interpretación de E es un objeto en D_α, el dominio de objetos del tipo α.

5.2. Estructura general de las reglas para razonamiento monótono

Una función F que preserva monotonía se puede representar de la siguiente manera:

$$\frac{X \Longrightarrow Y}{F(X) \Longrightarrow F(Y)}\ F\uparrow$$

Aquí se asume que X y Y son expresiones del tipo lógico α que está ordenado parcialmente mediante $\Longrightarrow$, que $F(X)$ y $F(Y)$ son expresiones del tipo β que está ordenado parcialmente mediante $\Longrightarrow$, y que F es una función que preserva el orden de tipo $\alpha \Longrightarrow \beta$.

Una forma de leer la regla es como una explicación del hecho de que F preserva el orden (monótona creciente). Otra manera de leer la regla es como una regla de inferencia disparada por una función F que se sabe que preserva el orden. $F\uparrow$ expresa que F preserva el orden.

Si la función F invierte el orden tenemos la siguiente regla:

$$\frac{X \Longrightarrow Y}{F(Y) \Longrightarrow F(X)}\ F\downarrow$$

Nuevamente, hay varias maneras de leer esta regla. $F\downarrow$ expresa que F invierte el orden (o que es monótona decreciente).

Para apreciar la generalidad de la regla de monotonía, veamos algunos casos especiales. Si $X, Y, F(X)$ y $F(Y)$ tienen tipo t, entonces $\Longrightarrow$ es consecuencia lógica (o implicación lógica), y $F(X)$ y $F(Y)$ son oraciones, de esta manera obtenemos:

$$\frac{X \Longrightarrow Y \quad F(X)}{F(Y)}\ F\uparrow$$

Un ejemplo de aplicación de esta regla es: inferir de "*Mary dances implies Mary moves*" (cuando tomamos "*Mary dances*" como X y "*Mary moves*" como Y) y "*Mary dances gracefully*" (con "*gracefully*" como F) que "*Mary moves gracefully*".

Para el caso de inversión de orden tenemos:

$$\frac{X \Longrightarrow Y \quad F(Y)}{F(X)}\ F\downarrow$$

Con X y Y como antes y leyendo F como negación, tenemos el siguiente ejemplo de esta regla: inferir de "*Mary dances implies Mary moves*" y "*Mary does not move*" (con "*does not move*" como F) que "*Mary does not dance*".

En el caso de que X y Y son conjuntos (con tipo $e \to t$) y $F(X)$ y $F(y)$ son valores veritativos, F tiene tipo $(e \to t) \to t$ (el tipo de los cuantificadores), obtenemos:

$$\frac{Q(X) \quad X \subseteq Y}{Q(Y)}\ Q\uparrow$$

Como ejemplo, que X sea "*dancing*", que Y sea "*moving*" y que Q sea "*everyone*". Entonces la regla dice que podemos concluir de "*everyone is dancing*" y "*dancing involves moving*" que "*everyone is moving*".

$$\frac{Q(Y) \quad X \subseteq Y}{Q(X)}\ Q\downarrow$$

Para este caso, que X sea "*dancing*", que Y sea "*moving*" y que Q sea "*nobody*". Entonces la regla dice que podemos concluir de "*nobody is moving*" y "*dancing involves moving*" que "*nobody is dancing*".

En efecto, F puede tener más estructura interna, es decir, $F(X)$ puede tener la forma de un cuantificador generalizado binario $Quant(X, P)$ o $Quant(P, X)$. Lo cual nos da cuatro posibles reglas de monotonía para cuantificadores binarios. Ejemplos de cuantificadores binarios son: *all*, con propiedades de monotonía $(\downarrow, \uparrow)$; *some*, con propiedades $(\uparrow, \uparrow)$; *no*, con $(\downarrow, \downarrow)$ y *most*, con $(-, \uparrow)$.

$$\frac{Quant(X, P) \quad X \subseteq Y}{Quant(Y, P)}\ Quant(\uparrow, -)$$

Ejemplo, infiera de "*some philosophers are mortal*" y "*philosophers are humans*" que "*some humans are mortal*".

$$\frac{Quant(P, X) \quad X \subseteq Y}{Quant(P, Y)}\ Quant(-, \uparrow)$$

Ejemplo, infiera de "*most philosophers are human*" y "*humans are mortal*" que "*most philosophers are mortal*".

$$\frac{Quant(Y, P) \quad X \subseteq Y}{Quant(X, P)}\ Quant(\downarrow, -)$$

Ejemplo, infiera de "*all humans are mortal*" y "*philosophers are human*" que "*no philosophers are mortal*".

$$\frac{Quant(P, Y) \quad X \subseteq Y}{Quant(P, X)}\ Quant(-, \downarrow)$$

Ejemplo, infiera de "*no philosophers are mortal*" y "*humans are mortal*" que "*no philosophers are human*".

6. Conclusiones

Es importante ofrecer un mejor balance entre precisión y cobertura en el reconocimiento de implicación textual, para ello creemos que un camino por explorar es aquel que le dé a la lógica natural la posibilidad de manejar la anáfora. Es decir, fusionar la lógica natural con alguna forma de lógica dinámica.

Referencias

1. Jurafsky, D., Martin, J.H.: Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Second edn. Pearson Education Inc. (2009)
2. Bos, J.: Is there a place for logic in recognizing textual entailment? Linguistic Issues in Language Technology **9**(3) (July 2013)
3. Burch, J., Clarke, E.M., McMillan, K.L.: Symbolic model checking: 10^{20} states and beyond. Information and Computation **98**(2) (1992) 142–170
4. Koller, A., Pinkal*, M.: Semantic research in computational linguistics. In Maienborn, C., von Heusinger, K., Portner, P., eds.: Semantics: An International Handbook of Natural Language Meaning. HSK Handbooks of Linguistics and Communication Science Series. Mouton de Gruyter (2012)
5. Androutsopoulos, I., Malakasiotis, P.: A survey of paraphrasing and textual entailment methods. Journal of Artificial Intelligence Research **38**(1) (2010) 135–187
6. Bos, J., Markert, K.: Recognising textual entailment with robust logical inference. In: MLCW 2005, volume LNAI 3944. (2006) 404–426
7. MacCartney, B., Manning, C.D.: Natural logic for textual inference. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Association for Computational Linguistics (June 2007) 193–200
8. Kamp, H., Reyle, U.: FROM DISCOURSE TO LOGIC-Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory. Kluwer Academic Publishers (1993)
9. Steedman, M., Baldridge, J.: Combinatory categorial grammar. In Borsley, R., Borjars, K., eds.: Non-Transformational Syntax: Formal and Explicit Models of Grammar. Wiley-Blackwell (2011)
10. Dekker, P.: A guide to dynamic semantics. Technical report, ILLC/Department of Philosophy, University of Amsterdam (2008)
11. van Eijck, J.: Natural logic for natural language. In ten Cate, B., Zeevat, H., eds.: 6th International Tbilisi Symposium on Logic, Language, and Computation Batumi, Georgia. Springer (2007) 216–230
12. Kamp, H., Van Genabith, J., Reyle, U.: Discourse representation theory. In Gabbay, D.M., Guenthner, F., eds.: Handbook of Philosophical Logic. Volume 15 of Handbook of Philosophical Logic. Springer Netherlands (2011) 125–394
13. Muskens, R.: Combining montague semantics and discourse representation. Linguistics and Philosophy **19** (1996) 143–186