

FUNDAÇÃO GETULIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA - FGV/EMAp
CURSO DE GRADUAÇÃO EM MATEMÁTICA APLICADA

Semântica Computacional para Textos Normativos

por

Guilherme Paulino Passos

Rio de Janeiro

2016

FUNDAÇÃO GETULIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA - FGV/EMAp
CURSO DE GRADUAÇÃO EM MATEMÁTICA APLICADA

Semântica Computacional para Textos Normativos

”Declaro ser o único autor do presente projeto de monografia que refere-se ao plano de trabalho a ser executado para continuidade da monografia e ressalto que não recorri a qualquer forma de colaboração ou auxílio de terceiros para realizá-lo a não ser nos casos e para os fins autorizados pelo professor orientador”

Guilherme Paulino Passos

Orientador: Prof. Dr. Alexandre Rademaker

Rio de Janeiro

2015

GUILHERME PAULINO PASSOS

Semântica Computacional para Textos Normativos

“Monografia apresentada à Escola de Matemática Aplicada - FGV/EMAp como requisito parcial para a obtenção do grau de Bacharel em Matemática Aplicada.”

Aprovado em ____ de _____ de ____ .

Grau atribuído à Monografia: ____ .

Professor Orientador: Prof. Dr. Alexandre Rademaker

Escola de Matemática Aplicada

Fundação Getulio Vargas

Professor Tutor: Prof. Dr. Paulo Cezar Pinto Carvalho

Escola de Matemática Aplicada

Fundação Getulio Vargas

Contents

1	Introdução	4
2	Representação semântica	5
2.1	Introdução	5
2.2	Cálculo Lambda	5
2.3	Armazenamento de Cooper	7
2.4	Armazenamento de Keller	7
2.5	<i>Hole Semantics</i>	7
3	...	8
4	Conclusão	9
5	References	10

1 Introdução

Não esquecer
de dizer que
linguagem-
objeto natural
será o inglês

2 Representação semântica

2.1 Introdução

Desejamos associar a cada expressão de linguagem natural um significado formal, simbólico. Além disso, desejamos fazê-lo de modo algorítmico, que possa ser reproduzido por um computador.

A linguagem formal que utilizaremos para representar o significado de frases é *lógica de primeira ordem*. Jurafsky and Martin (2009) apresentam como propriedades interessantes para representações: verificabilidade, representações não ambíguas, existência de uma forma canônica, capacidade de inferência e uso de variáveis e expressividade. Todas estas são possuídas pela lógica de primeira ordem. Também uma interessante propriedade da lógica de primeira ordem é sua relativa intuitividade. Bastando explicar o que significam os símbolos conectivos (como \wedge (significando “e”) e \rightarrow (significando “se ... então ... ”)), bem como os quantificadores (\forall (“para todo”) e \exists (“existe”), uma expressão formal em lógica é compreensível.

Conferir se falo de lógica aqui ou se puxo isso para a introdução.

é verdade?

Ainda que tenhamos escolhido a lógica de primeira ordem para ser a linguagem das representações semânticas para frases, isto não nos informa qual deve ser a representação semântica de palavras e expressões menores. Talvez algumas poderiam ser feitas por termos, mas não está de todo claro qual seria o significado de uma expressão como “to run” (“correr”) ou “that walks” (“que anda”).

Em nossos pressupostos, adotamos o *Princípio da Composicionalidade*. Segundo o mesmo, o significado de expressões complexas é função das expressões mais simples que a compõem. Em um exemplo como “Caim kills Abel”, isto nos informa que o significado desta frase depende do significado de “Caim”, “kills” e “Abel”. Entretanto, isto não nos diz como funciona esta dependência, ou a função que leva o significado das expressões simples ao da expressão complexa.

Por exemplo, podemos entender que o significado de “kills” é o predicado binário `kill(..., ...)`, onde convencionamos que o primeiro argumento é o agressor (isto é, aquele que mata) e o segundo argumento é a vítima (aquele que é morto). Também podemos entender os significados de “Caim” e “Abel” como as constantes `caim` e `abel`, respectivamente. Assim, apesar de `kill(abel, caim)` ser formada com o significado destes três termos, respeitando a composicionalidade, esta não é a expressão que queremos, e

`sim kill(caim, abel).`

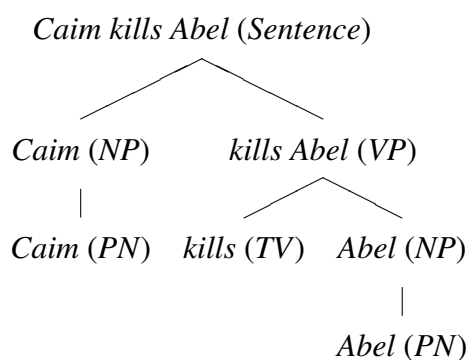
O que nos falta é a *sintaxe*. A sintaxe é o conjunto de regras e processos que organizam a estrutura de frases.

achar uma boa referência

Assim, as palavras em uma frase existem em relação a uma certa estrutura, que é essencial para capturar o significado. No inglês, com a estrutura *Sujeito - Verbo - Predicado*, entendemos que “*Caim kills Abel*” significa `kill(caim, abel)`, e não `kill(abel, caim)`.

O foco deste trabalho não é na sintaxe, de modo que utilizamos uma sintaxe simples: a gramática é implementada pelo mecanismo de Gramática de Cláusulas Definidas (*Definite Clause Grammar* - DCG). A análise sintática é feita na forma de uma árvore cujos nós que são folhas são categorias sintáticas básicas (tais como sujeito (*noun*), verbo transitivo (*transitive verb*) e quantificador (*quantifier*, considerado caso particular de *determiner*). Já os nós que não são folhas representam categorias sintáticas complexas (tais como sintagma nominal (*noun phrase*) ou sintagma verbal (*verb phrase*). (Blackburn and Bos, 2005, p. 58)

Um exemplo de tal árvore, para a frase “*Caim kills Abel*”, seria:



Aqui, temos as classes sintáticas:

NP – *noun phrase* (sintagma nominal)

PN – *proper noun* (nome próprio)

VP – *verb phrase* (sintagma verbal)

TV – *transitive verb* (verbo transitivo)

2.2 Cálculo Lambda

2.3 Armazenamento de Cooper

2.4 Armazenamento de Keller

2.5 *Hole Semantics*

3 ...

4 Conclusão

5 References

Patrick Blackburn and Johan Bos. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI, 2005.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009. ISBN 0131873210.