



**Guilherme Sant'Anna Varela**

**Anotação Profunda de Papéis Semânticos para  
o Português**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática da PUC-Rio.

Orientador: Prof. Sérgio Colcher

Rio de Janeiro  
Janeiro de 2019



**Guilherme Sant'Anna Varela**

**Anotação Profunda de Papéis Semânticos para  
o Português**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Sérgio Colcher**

Orientador

Departamento de Informática – PUC-Rio

**Prof<sup>a</sup>. Maria Claudia de Freitas**

Departamento de Letras – PUC-Rio

**Prof. Leandro Guimarães Marques Alvim**

– UFRRJ

**Prof. Ruy Luiz Milidiú**

Departamento de Informática – PUC-Rio

**Prof. Márcio da Silveira Carvalho**

Coordenador Setorial do Centro

Técnico Científico Departamento de Informática – PUC-Rio

Rio de Janeiro, 21 de Janeiro de 2019

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Guilherme Sant'Anna Varela**

Graduated in electrical engineering by the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). His research is focused in Machine Learning and Natural Language Processing.

#### Ficha Catalográfica

Varela, Guilherme Sant'Anna

Anotação Profunda de Papéis Semânticos para o Português / Guilherme Sant'Anna Varela; orientador: Sérgio Colcher. – 2019.

74 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2019.

Inclui bibliografia

1. Informática – Teses. 2. Aprendizado de Máquina;. 3. Aprendizado de Máquina Profundo;. 4. Processamento de Linguagem Natural;. 5. Anotação de Papéis Semânticos;. I. Colcher, Sérgio. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

## Agradecimentos

O primeiro obrigado vai para o meu orientador, Prof. Ruy Luiz Milidiú com quem a convivência e as conversas geraram contribuições tanto profissionais como pessoais. Sou grato também pela revisão minuciosa da Prof<sup>a</sup> Maria Claudia de Freitas, que enriqueceram esse trabalho ( os eventuais erros residuais são de autoria minha). Ao Prof. Sérgio Colcher obrigado pelo apoio nessa jornada.

Obrigado à PUC-Rio que me proporcionou essa oportunidade de crescimento pessoal.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Aos meus amigos Rafael Rocha e Luís Müller, obrigado pelas conversas construtivas sobre aprendizado de máquina. Ao Miguel Brito, rato de laboratório, um abraço. Ao time do Digibot agradeço pela camaradagem infinita.

À diretoria do AML é uma tremenda honra trabalhar com vocês.

À minha família agradeço o apoio.

À minha noiva, nós conseguimos.

## Resumo

Varela, Guilherme Sant'Anna; Colcher, Sérgio. **Anotação Profunda de Papéis Semânticos para o Português**. Rio de Janeiro, 2019. 74p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Vivemos em um mundo complexo, no qual incontáveis fatores aparentemente desconexos – tais como a lei de Moore que dita um aumento exponencial da capacidade de processamento em um *chip* de silício, a queda do custo de espaço de armazenamento e a adoção em massa de *smartphones* colaboram para a formação de uma sociedade progressivamente interdependente. Todos os dias são criados 2,5 quintilhões de *bytes* de dados, de fato 90% dos dados no mundo foram criados nos últimos dois anos. Domar os padrões salientes aos dados separando informação do caos torna-se uma necessidade iminente para a tomada de decisão dos indivíduos e para sobrevivência de organizações. Nesse cenário a melhor resposta dos pesquisadores de Processamento de Linguagem Natural encontra-se na tarefa de Anotação de Papéis Semânticos. APS é a tarefa que tem o audacioso objetivo de compreender eventos, buscando determinar ‘Quem fez o que e aonde’, ‘Quais foram os beneficiados?’ ou ‘Qual o meio utilizado para atingir os fins’. APS serve como tarefa intermediária para várias aplicações de alto nível e.g. *information extraction*, *question and answering* e agentes conversacionais. Tradicionalmente, resultados satisfatórios eram obtidos apenas com alta dependência de conhecimento específico de domínio. Para o português, através desta abordagem, o sistema estado da arte da tarefa para é de 79,6% de pontuação F1. Sistemas mais recentes dependem de uma série de subtarefas, obtém 58% de pontuação F1. Nessa dissertação, exploramos um novo paradigma utilizando redes neurais recorrentes, para o idioma do português do Brasil, e sem subtarefas intermediárias obtendo uma pontuação de 66,23

## Palavras-chave

Aprendizado de Máquina; Aprendizado de Máquina Profundo; Processamento de Linguagem Natural; Anotação de Papéis Semânticos;

## Abstract

Varela, Guilherme Sant'Anna; Colcher, Sérgio (Advisor). **Deep Semantic Role Labeling for Portuguese**. Rio de Janeiro, 2019. 74p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

We live in a complex world in which a myriad of seemingly unrelated factors – such as Moore’s law which states that the processing capacity on a silicon wafer should increase exponentially, the fall of storage costs and mass adoption of smart-phones contribute to the formation of an increasingly inter-dependent society: 2.5 quintillion bytes of data are generated every day, in fact ninety percent of the world’s data were created in the last few years. Harnessing the emerging patterns within the data, effectively separating information from chaos is crucial for both individual decision making as well as for the survival of organizations. In this scenario the best answer from Natural Language Processing researchers is the task of Semantic Role Labeling. SRL is the task the concerns itself with the audacious goal of event understanding, which means determining ‘Who did what to whom’, ‘Who was the beneficiary?’ or ‘What were the means to achieve some goal’. APS is also an intermediary task to high level applications such as information extraction, question and answering and chatbots. Traditionally, satisfactory results were obtained only by the introduction of highly specific domain knowledge. For Portuguese, this approach is able to yields a F1 score of 79.6%. Recent systems, rely on a pipeline of sub-tasks, yielding a F1 score of 58%. In this dissertation, we adopt a new paradigm using recurrent neural networks for the Brazilian Portuguese, that does not rely on a pipeline, our system obtains a score of 66.23%.

## Keywords

Machine Learning; Deep Learning; Natural Language Processing; Semantic Role Labeling;

## Sumário

1	Introdução	12
1.1	Contextualização	12
1.2	Objetivo	13
1.3	Organização	14
2	Anotação Automática de Papéis Semânticos	15
2.1	Score $F_1$	15
2.2	Decomposição em Subtarefas	16
2.2.1	Poda de Árvore Sintagmática	16
2.2.2	Identificação de Argumentos	17
2.2.3	Classificação de Argumentos	18
2.2.4	Inferência	18
2.2.5	Reconhecimento de predicado alvo	19
2.3	Trabalhos Relacionados	19
2.3.1	Idioma: Inglês	19
2.3.2	Idioma: Português	24
3	Anotação de Papéis Semânticos	26
3.1	Teoria Lingüística	26
3.2	PropBank	29
3.3	PropBank.Br	32
4	Aprendendo Seqüências	36
4.1	Representação de Palavras	36
4.1.1	Word2Vec	38
4.1.1.1	CBOW: Sacola de palavras contínua	38
4.1.1.2	Skipgram	39
4.1.2	Wang2Vec	39
4.1.3	GloVe	39
4.2	Redes Neurais Recorrentes	40
4.2.1	Redes Neurais Recorrentes Bidirecionais	41
4.2.2	O problema do gradiente minguante explosivo	42
4.2.3	Long Short Term Memory	43
4.3	Predição Estruturada	44
4.3.1	Conditional Random Fields	45
4.3.2	Perceptron Estruturado	47
5	Metodologia	53
5.1	Pré processamento	53
5.1.1	Seleção de atributos padrão ouro	53
5.1.2	Transformação de <i>tokens</i>	54
5.1.3	Extração de atributos	55
5.2	Redes Neurais Recorrentes	56
5.2.1	BiLSTM	56

5.2.2	DBLSTM	57
5.3	Predição Estruturada	57
6	Resultados Experimentais	<b>59</b>
6.1	Etapa 1	59
6.2	Etapa 2	61
7	Conclusão	<b>64</b>
8	Referências bibliográficas	<b>65</b>
A	Apêndice	<b>73</b>
A.1	O Grafo de Computação	73

## Lista de figuras

Figura 2.1	Sintaxe e Semântica (fonte: <i>PropBank.Br</i> )	17
Figura 2.2	Rede Convolutiva (28).	21
Figura 2.3	Atributos no sistema (9).	21
Figura 2.4	Sentença segundo um mecanismo de atenção (33).	23
Figura 2.5	Seqüência de subtarefas (22).	25
Figura 3.1	Classe de verbo <i>spray</i> .	28
Figura 3.2	Três proposições do <i>PropBank</i>	32
Figura 3.3	Exemplo do PropBank.Br	34
Figura 4.1	<i>word2vec</i> (53)	37
Figura 4.2	CBOW: sacola contínua de palavras (53)	38
Figura 4.3	<i>Skipgram</i> (53)	39
Figura 4.4	O grafo de computação de uma RNR	40
Figura 4.5	O grafo de computação de uma Bi RNR	41
Figura 4.6	O grafo de computação de uma LSTM	43
Figura 4.7	O <i>perceptron</i> (62)	47
Figura 4.8	Treliça	48
Figura 4.9	Treliça	50
Figura 4.10	Mapa de custo $t = 0$	50
Figura 4.11	Mapa de custo $t = 1$	51
Figura 4.12	Mapa de custo $t = 2$	51
Figura 4.13	Mapa de custo $t = 2$	51
Figura 4.14	Mapa de custo $t = 4$	52
Figura 5.1	Pré processamento 1 <sup>a</sup> etapa.	54
Figura 5.2	Pré processamento 2 <sup>a</sup> etapa.	55
Figura 5.3	Pré processamento 3 <sup>a</sup> etapa.	56
Figura 5.4	BiLSTM Diagrama	57
Figura 5.5	DBLSTM Diagrama	58
Figura A.1	O gráfico de computação.	73
Figura A.2	Desdobramento do grafo de computação.	74

## Lista de tabelas

Tabela 2.1	<i>Leaderboard</i> para sistema de APS para o inglês.	23
Tabela 2.2	<i>Leaderboard</i> para sistema de APS para o português.	24
Tabela 3.1	Os Proto-Papéis de Dowty	28
Tabela 3.2	Descrição dos argumentos dos verbos comprar e vender	30
Tabela 3.3	Papéis usualmente assumidos pelos argumentos	30
Tabela 3.4	Papéis adjuntos (45)	30
Tabela 3.5	PropBank.Br v1.0 partições.	35
Tabela 5.1	<i>Part-of-Speech</i>	54
Tabela 6.1	Resultados da experimentais 1.	60
Tabela 6.2	Resultados etapa 2.	62
Tabela 6.3	Resultados finais.	63
Tabela 6.4	Sistemas por atributos padrão ouro	63
Tabela 6.5	Sistemas por sub tarefas	63

## Lista de Abreviaturas

APS – Anotação de Papéis Semânticos

AAPS – Anotação Automática de Papéis Semânticos

PLN – Processamento de Linguagem Natural

RN – Redes Neurais (Artificiais)

MLP – Multi Layer Perceptron

SVM – *Support Vector Machines*

RNR – Redes Neurais Recorrentes

LSTM – *Long Short-Term Memory Networks*

PGME – Problema do Gradiente Minguante/Explosivo

# 1 Introdução

## 1.1 Contextualização

A análise de semântica do texto, em nível da sentença, é voltada para a caracterização de eventos tais quais determinar ‘quem’ fez ‘oque’ à ‘quem’, ‘aonde’, ‘como’ e ‘quando’. O predicado de uma oração (tipicamente o verbo) estabelece ‘oque’ ocorreu e os demais constituintes da sentença expressam participantes desse evento tais como (‘como’ e ‘quando’). É prerrogativa da **anotação de papéis semânticos** (APS) indicar exatamente quais são os relacionamentos semânticos entre esses participantes e propriedades, essas relações são limitadas à uma determinada lista pré especificada pelo predicado (ou classe de predicados). A fim de atingir esse objetivo, os argumentos constituintes que assumem algum papel precisam ser primeiramente identificados e seus papéis devem ser devidamente alocados. (5) A tarefa possui a dupla atribuição: Identificação dos constituintes que possuem um papel semântico, inclusive aqueles que não possuem papel algum, e a conseqüente classificação destes papéis. Conforme (1) (72):

- [João<sub>AGENTE</sub>] [quebrou<sub>V</sub>] [a janela<sub>PACIENTE</sub>] [com a bola<sub>INSTRUMENTO</sub>].
- [Os turistas<sub>EXPERIENCIADOR</sub>] [admiraram<sub>V</sub>] [a paisagem<sub>TEMA</sub>].

A tarefa de anotação automática de papéis semânticos foi criada em 2002 (24) quando pesquisadores utilizaram o recurso lexical disponível chamado FrameNet (3) para fazer anotação de papéis específicos tais como vendedor, comprador e orador. A pesquisa no entanto seria propulsionada apenas em 2004 e 2005 quando a tarefa de anotação automática de papéis semânticos, se tornou tema da *ConLL Shared Task 2004* e *ConLL Shared Task 2005* respectivamente. As competições tornaram o recurso lexical PropBank (43) disponível para pesquisadores. O PropBank diferentemente do FrameNet é o primeiro recurso lexical produzido com o objetivo explícito de servir de *corpus* para a pesquisa em Processamento de Linguagem Natural. E desde o seu lançamento outros recursos lexicais têm sido produzidos com esse mesmo objetivo; OntoNotes (73), NomBank (74). Em múltiplos idiomas; Português (47),

Catalão, Chinês, Czeko, Alemão, Japonês e Espanhol (76). E por meio de diferentes competições; *ConLL Shared Task 2005* (45), *SemEval 2007 Task 9* (76), *ConLL Shared Task 2012* (73).

Anotação automática de papéis semânticos serve como tarefa intermediária para uma série de aplicações tais como extração de informações (10, 11), sistemas de pergunta resposta (12), resumo de texto (13) e tradução automática (14, 15, 16). Tais aplicações são orientadas à prover capacidade interpretativa e consequente exploração da progressiva geração de dados<sup>1</sup> por parte da humanidade. A tarefa é também desafiadora, tendo no espaço de uma década modesta melhora da pontuação  $F_1$  1,9% partindo de 79,4% (8) em 2005 para 82,8% (9) em 2015. A pontuação estado da arte para o idioma Inglês é  $F_1$  86,4% (32).

O recurso lexical do PropBank.Br (47) segue os padrões de anotação estabelecidos pelo PropBank, e embora tenha recebido atualizações a versão mais utilizada por pesquisadores de anotação de papéis semânticos para o Português é a 1.0. Os trabalhos para o Português seguem abordagens semelhantes à literatura para o Inglês, o primeiro trabalho foi produzido por Alva Machego, 2013 (20) baseado em (8). O trabalho faz amplo uso de atributos sintático tais como árvores sintagmáticas padrão ouro e é fruto de uma sequência de subtarefas. Tendo obtido uma pontuação  $F_1$  79,7%. O segundo trabalho, foi produzido por Fonseca, 2013 (41) baseado em (28), e utiliza uma rede neural convolutiva, obtendo uma pontuação  $F_1$  57,25%. O terceiro trabalho, foi produzido por Garrido, 2017 (22), e utilizando uma sequência de subtarefas com atributos linguísticos estimados através de CRF. O trabalho obteve uma pontuação  $F_1$  58,00%.

## 1.2

### Objetivo

Esta dissertação propõe um modelo para anotação de papéis semânticos para o português do Brasil, utilizando o recurso lexical PropBank.Br. A arquitetura do modelo faz uso de redes neurais recorrentes, que são redes neurais multicamadas voltadas ao aprendizado de seqüências. A escolha de redes neurais recorrentes é inspirada no trabalho (9), que foi o primeiro no idioma do inglês a superar (8) mesmo sem utilizar sequência de subtarefas e apenas empregando o predicado padrão ouro. O objetivo do trabalho é desenvolver um sistema apenas com a informação do predicado padrão ouro, e produzir um sistema fim-a-fim para o português do Brasil.

<sup>1</sup><https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WRL12345USEN>

### 1.3 Organização

Este documento está organizado conforme descrito a seguir: O Capítulo 2 traz a evolução da tarefa de APS tanto para o inglês quanto para o português, avaliada de forma objetiva através da métrica  $F_1$  e de forma qualitativa abordando características como a prevalência de atributos linguísticos, ou se utiliza ou não múltiplas subtarefas para a resolução sua resolução. O Capítulo 3 apresenta a teoria lingüística que orientou o desenvolvimento do PropBank. Naturalmente, o PropBank também é apresentado assim como a sua variante no idioma do português do Brasil – o PropBank.Br. O Capítulo 4 traz os modelos de representação de texto *word2vec* assim como as arquiteturas de apredizado profundo que vêm sendo empregadas por pesquisadores para a resolução da tarefa. No Capítulo 5, apresentamos a arquitetura adotada para rodar os experimentos. O Capítulo 7 traz as conclusões deste trabalho, bem como recomendações para trabalhos futuros.

Este Capítulo é dedicado à exposição da evolução da tarefa de anotação de papéis semânticos conforme registrada na literatura, para os idiomas do Inglês e do Português.

A motivação é que, o volume de publicações para o Inglês é predominante, sendo assim os autores dos trabalhos em Português tendem a adaptar sistemas do Inglês para o Português. Alguns trabalhos para AAPS em Português como Hartmann e Alúcio, 2017 (17) e Amancio e Alúcio 2009 (18), não são diretamente comparáveis com o sistema apresentado já que utilizam um *corpus* diferente ou são dedicados a modelar apenas as etiquetas A0 e A1 do PropBank.Br.

Além do *score*  $F_1$  os sistemas, podem ser qualificados pela sua dependência de atributos linguísticos e naqueles que apresentam multitarefas. A adoção de arquiteturas de aprendizado profundo possibilitou, que os sistemas fossem treinados de maneira fim à fim, isto é sem a necessidade da resolução de subtarefas intermediárias que efetivamente limitam a pontuação  $F_1$  máxima. Enquanto, ao mesmo tempo, reduziu a dependência de atributos linguísticos.

## 2.1

### Score $F_1$

O *score*  $F_1$  é a principal métrica de avaliação de um sistema de anotação de papéis semânticos, ela é composta de precisão ( $p$ ) e cobertura ( $c$ ). Precisão é a proporção de argumentos preditos que estão corretas. Cobertura é a proporção de argumentos corretos que é predita pelo sistema.  $F_1$  é a média harmônica entre precisão e cobertura definida como:

$$F_1 = \frac{2pc}{p + c} \quad (2-1)$$

A métrica é calculada utilizando as predições do sistema em um conjunto de teste com exemplos não vistos durante a etapa de treino. No caso específico da *ConLL 2005 Shared Task* há dois conjuntos de teste que são avaliados conjuntamente. É o conjunto de teste WSJ seção 21 (*in-sample*) e um outro *corpus* anotado Brown. Autores tendem a divulgar os *scores* também de forma separada.

## 2.2

### Decomposição em Subtarefas

Os primeiros sistemas de anotação automática de papéis semânticos, são predominantemente dependentes de uma seqüência de tarefas que são executadas de forma serial, (8, 19, 20). Então os pesquisadores estavam concentrados em entender a dependência entre sintaxe e semântica, utilizando duas construções lingüísticas. As **árvores sintagmáticas** (*parse trees* ou *synthactic tree*) e sintagmas (*shallow parsing* ou *chunk*) são dois dos atributos lingüísticos que contribuem para a tarefa de APS. A Figura 2.1 ilustra um exemplo de árvore de sintágmata, e os respectivos papéis semânticos em diferentes formatos.

As arquiteturas eram compostas dos seguintes estágios: **Poda de Árvore de Sintagmas**, **Identificação de Argumentos**, **Classificação de Argumentos** e **Inferência**. Em particular o objetivo das etapas de poda e identificação é identificar candidatos para um dado predicado. Nos primeiros três estágios decisões são feitas de maneira independente, i.e informações entre argumentos não é incorporada e são combinadas no estágio de inferência, com a introdução de restrições lingüísticas (27). Adicionalmente, sistemas que processam texto bruto implementam a tarefa de **Identificação de predicado** com o objetivo de identificar o verbo que faz o papel de predicado na sentença.

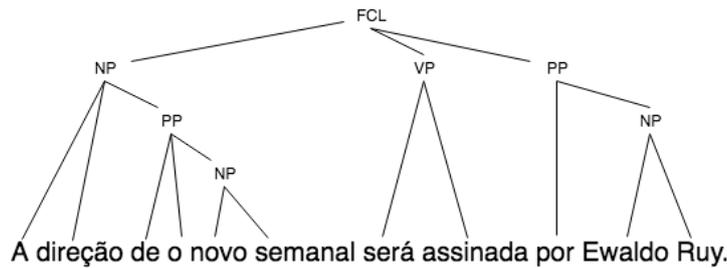
Trabalhos mais recentes têm empregado algumas dessas tarefas como He, *et al* 2017 (21) utilizam a etapa de inferência com um algoritmo the menor caminho **A\***. Enquanto Garrido, 2017 (22) fazem identificação de argumentos, classificação de argumentos e identificação de predicados.

#### 2.2.1

##### Poda de Árvore Sintagmática

O objetivo da poda é excluir constituintes da árvore com poucas chances de se tornar argumento. Trata-se de um algoritmo recursivo em dois passos Xue e Palmer, 2004 (23). O algoritmo seleciona os constituintes vizinhos ao verbo, caso o tipo de constituinte seja **PP** os filhos também são retornados. O passo recursivo faz com que o próximo nó seja o pai do constituinte **VP**, o algoritmo se encerra quando chega à raiz. No caso do exemplo na Figura 2.1 os candidatos à argumentos são: [A direção de o novo semanal<sub>NP</sub>], [por Ewaldo Ruy<sub>PP</sub>], [Ewaldo Ruy<sub>NP</sub>]

Árvore de Sintáguas:



Sintáguas:

[A direção<sub>NP</sub>] [de o<sub>PP</sub>] [novo semanal<sub>NP</sub>] [será assinada<sub>V</sub>] [por<sub>PP</sub>] [Ewaldo Ruy<sub>NP</sub>].

APS (arvore plana):

[A direção de o novo semanal<sub>A1</sub>] [será assinada<sub>V</sub>] [por Ewaldo Ruy<sub>A0</sub>].

APS (IOB):

B-A1 I-A1 I-A1 I-A1 I-A1 I-A1 B-V I-V B-A0 I-A0 I-A0 O

APS (IOBES):

B-A1 I-A1 I-A1 I-A1 I-A1 E-A1 B-V E-V B-A0 I-A0 E-A0 O

Figura 2.1: Sintaxe e Semântica (fonte: *PropBank.Br*)

## 2.2.2

### Identificação de Argumentos

Nesta fase os constituintes que participam de um argumento para o predicado alvo são delimitados. Quando existe a árvore sintática os sintagmas gerados no passo anterior devem ser classificados por um classificador binário que gera os candidatos. No caso do exemplo na Figura 2.1 a saída desta etapa seria [A direção de o novo semanal<sub>1</sub>], [por Ewaldo Ruy<sub>1</sub>], [Ewaldo Ruy<sub>0</sub>].

Quando apenas *chunks* estão disponíveis então os classificadores devem discriminar entre início e final de intervalo. Para representar intervalos a partir *tokens* são utilizados IOB e IOBES.

1. **IOB** *Tokens* que marcam o início de um argumento são marcados com a etiqueta B. Demais *tokens* participando de um argumento são marcados com a etiqueta I. Se um *token* não participa de um argumento para determinado predicado então serão marcados com a etiqueta O.
2. **IOBES** A principal distinção está no tratamento de argumentos maiores que um *token*. Para tais intervalos, a etiqueta B marca o início do argumento, a etiqueta E marca o final do argumento, já os *tokens* intermediários são marcados com a etiqueta I. Se um argumento tem

tamanho um então recebe a etiqueta *S* e finalmente se um *token* não é argumento este recebe etiqueta *O*.

Uma representação da saída da etapa de indentificação de argumentos seria:

IOB:

A<sub>B</sub> direção<sub>I</sub> de<sub>I</sub> o<sub>I</sub> novo<sub>I</sub> semanal<sub>I</sub> será<sub>B</sub> assinada<sub>I</sub> por<sub>B</sub> Ewaldo<sub>I</sub> Ruy<sub>I</sub> .o

IOBES:

A<sub>B</sub> direção<sub>I</sub> de<sub>I</sub> o<sub>I</sub> novo<sub>I</sub> semanal<sub>E</sub> será<sub>B</sub> assinada<sub>E</sub> por<sub>B</sub> Ewaldo<sub>I</sub> Ruy<sub>E</sub> .o

### 2.2.3

#### Classificação de Argumentos

Nesta fase, os argumentos delimitados providos da etapa de identificação de argumentos são classificados. Uma etiqueta *NULL* ou *O* adicionada para denominar candidatos a argumentos que não são válidos (no caso é o ponto final).

IOB:

A<sub>A1</sub> direção<sub>A1</sub> de<sub>A1</sub> o<sub>A1</sub> novo<sub>A1</sub> semanal<sub>A1</sub> será<sub>V</sub> assinada<sub>V</sub> por<sub>A0</sub> Ewaldo<sub>A0</sub> Ruy<sub>A0</sub> .NULL

IOBES:

A<sub>A1</sub> direção<sub>A1</sub> de<sub>A1</sub> o<sub>A1</sub> novo<sub>A1</sub> semanal<sub>A1</sub> será<sub>V</sub> assinada<sub>V</sub> por<sub>A0</sub> Ewaldo<sub>A0</sub> Ruy<sub>A0</sub> .NULL

### 2.2.4

#### Inferência

Na fase de inferência, regras lingüísticas e de consistência são utilizadas para unir as saídas das etapas anteriores (27) (21), este processo garante um formato válido de saída. Entre as restrições utilizadas estão:

- Argumentos não podem se sobrepor ao predicado.
- Argumentos devem estar contidos em uma oração.
- Se um predicado está fora da oração, seus argumentos não podem estar dentro do escopo desta oração.
- Classes de argumento A0–A5 não podem se repetir.
- Dado um predicado, algumas classes de argumentos são proibidas por exemplo o verbo ‘escrever’ não aceita argumento A5.

### 2.2.5

#### Reconhecimento de predicado alvo

Trata-se de um classificador binário que associa à cada *token* se ele é ou não predicado para alguma sentença. Nas competições da *CoNLL 2004 Shared Task* e *CoNLL 2005 Shared Task* esta etapa é considerada como resolvida e o predicado é desconsiderado durante o computo da métrica  $F_1$  para a competição.

A<sub>0</sub> direção<sub>0</sub> de<sub>0</sub> o<sub>0</sub> novo<sub>0</sub> semanal<sub>0</sub> será<sub>1</sub> assinada<sub>1</sub> por<sub>0</sub> Ewaldo<sub>0</sub> Ruy<sub>1</sub> .<sub>0</sub>

## 2.3

### Trabalhos Relacionados

#### 2.3.1

##### Idioma: Inglês

Em 2002, Gildea and Jurafksy (24) desenvolveram o primeiro sistema para anotar de forma automática papéis para 50.000 sentenças em um *corpus* manufaturado em acordo com o formato estabelecido no FrameNet (25). O sistema foi capaz de determinar **Agente** e **Tema** de uma sentença e alguns predicados específicos como **Orador** e **Mensagem**.

Em 2004, AAPS foi o tema da conferência *CoNLL 2004 Shared Task*, onde seis seções do *Wall Street Journal* que é parte do *Penn Treebank* (26) foram anotadas com os papéis semânticos, as seções 15-18 foram utilizadas como treino, seção 20 foi reservada para desenvolvimento e finalmente a seção 21 foi utilizada como teste (51). O sistema prevalente à época, Hacioglu, *et al* 2004 (19) desenvolve um anotador sintático (e.g *chunks* NP, PP, VP), que por sua vez são reclassificados como argumentos semânticos (e.g B-A0, I-A1), em formato IOB. O anotador usa um separador SVM<sup>1</sup> e obteve desempenho na medida de  $F_1$  69.49%.

Em 2005, *CoNLL 2005 Shared Task* também foi dedicada à AAPS, em comparação com o ano anterior teve a introdução de anotação de árvores sintagmáticas, aumento no número de exemplos e avaliação cruzada no Brown *corpus*. As seções WSJ 02-21 foram utilizadas como treino, 24 para desenvolvimento e 23 para teste. E mais as três seções no Brown *corpus* anotadas utilizando as regras do PropBank (45). O anotador é composto por uma *pipeline* de quatro estágios: (i) poda: apenas constituintes da árvore sintagmática são considerados como argumentos e uma heurística é utilizada para podar

<sup>1</sup>SVM *Support Vector Machines*

candidatos pouco prováveis. (ii) Identificação de argumentos: utiliza um classificador binário para identificar se um candidato é ou não um argumento. (iii) Classificação de argumentos: aos argumentos identificados na etapa anterior, o anotador associa uma classe. (iv) Inferência: nesse estágio restrições lingüísticas (e.g ‘cada verbo aceita apenas um argumento’ e ‘argumentos não se sobrepõe’) são aplicadas de forma a combinar argumentos. O sistema obteve um *score*  $F_1$  79,44% no conjunto de teste oficial WSJ e um *score*  $F_1$  67,75% no conjunto de extra oficial Brown para um total combinado de *score*  $F_1$  77,92% WSJ + Brown. Alguns anos depois em 2008, Punyakanok, *et al* (27) afirmam que os atributos sintáticos são necessários para obter resultados satisfatórios nessa tarefa.

Em 2011, Collobert, *et al* (28) preconizou a aplicação de arquiteturas de aprendizado profundo no domínio de PLN, o sistema SENNA<sup>2</sup> é uma rede convolutiva multicamadas, entre as inovações introduzidas: (i) Modelo de linguagem: um modelo de linguagem foi treinado separadamente para capturar relacionamentos entre as palavras. (ii) Redes neurais convolutivas multicamadas em detrimento de uma *pipeline* (iii) Uma arquitetura, várias tarefas ao invés da composição de várias sub tarefas. (iv) Menor dependência de atributos lingüísticos, no caso de APS apenas árvores sintagmáticas foram utilizadas. A Figura 2.2 ilustra a arquitetura proposta, primeiro os atributos textuais passam por uma operação de *table lookup*, depois passam por uma operação de convolução e *max pooling* gerando filtros, a próxima camada combina os resultados e aplica uma função de ativação, a última camada produz uma função de perda em nível de seqüência. SENNA obteve estado da arte para as tarefas de *PoS tagging*, *NER* e *Chunking*. O mesmo não se aplicou a tarefa de Anotação de Papéis Semânticos e tendo obtido um  $F_1$  75,49% no conjunto de teste oficial WSJ.

Em Zhou e Xu, 2015 (9) os autores propõe uma outra estrutura de aprendizado profundo, trata-se de um sistema de rede neural recorrente. Alimentado por proposições do *corpus*; as palavras são convertidas para vetores de tamanho 50, utilizando GloVe. e propagados pela RNR com 8 camadas de unidades LSTM ligadas de forma interlaçada. As representações são fornecidas para uma camada CRF, que então é treinada com uma representação das etiquetas em formato IOB. A Figura 2.3 ilustra um exemplo utilizado pelo sistema: (i) *time* significa a posição do *token* dentro da proposição. É reiniciado em 1 à cada novo exemplo. (ii) *argu* o *token* em si, (iii) *pred* o predicado padrão ouro, (iv) *ctx-p* uma janela de tamanho fixo em torno do predicado, que é obtida nesse exemplo coletando *tokens* precedentes e sucessores *been* e

<sup>2</sup>*Semantic/syntactic Extraction using a Neural Network Architecture*

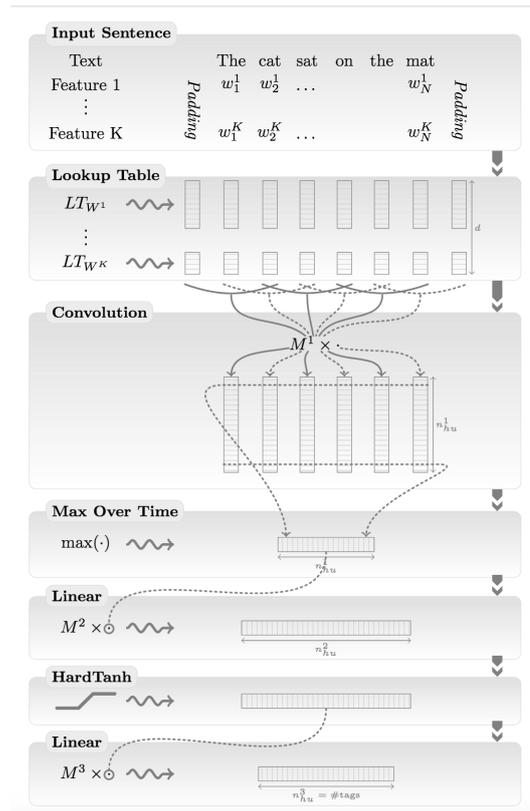


Figura 2.2: Rede Convolutiva (28).

set respectivamente. (v)  $m_r$  serve como um marcador da região do predicado, trata-se de uma variável booleana que assume 0 se o predicado não foi visto e 1 se o predicado já foi encontrado em um tempo anterior. A motivação para os atributos (iv) e (v) é apenas a informação do predicado não é suficiente para descrever toda a variação de sentidos que o mesmo pode assumir. Seu sistema obteve um *score*  $F_1$  81,07% no conjunto de teste oficial WSJ + Brown da *CoNLL 2005 Shared Task* estabelecendo o novo estado da arte. E reverteu o consenso estabelecido de que as árvores sintagmáticas são necessárias para haja um bom desempenho em AAPS.

time	argu	pred	ctx-p	$m_r$	label
1	A	set	been set .	0	B-A1
2	record	set	been set .	0	I-A1
3	date	set	been set .	0	I-A1
4	has	set	been set .	0	O
5	n't	set	been set .	0	B-AM-NEG
6	been	set	been set .	1	O
7	set	set	been set .	1	B-V
8	.	set	been set .	1	O

Figura 2.3: Atributos no sistema (9).

Em He, *et al* 2017 (21), os pesquisadores ainda utilizando redes neurais recorrentes, introduzem uma série de inovações; orientadas à minimização do tempo de treino como; (i) Simplificação da camada de saída; de CRF

para *Softmax*. Ao alívio do PGME: (ii) *Highway Connections* Srivastava, *et al* 2015 (29) Zhang, *et al* 2016 (30) (iii) *Dropout* Recorrente (31) E a técnicas estabelecidas no campo de PLN, voltadas à melhoria da qualidade de previsão; (iv) Decodificação com etiquetas BIO, com restrições lingüísticas (v) Introdução de comitê de especialistas. O sistema obteve um *score*  $F_1$  84,6% no conjunto de teste oficial WSJ e um *score*  $F_1$  73,6% no conjunto de extra oficial Brown para um total combinado de *score*  $F_1$  83,2% WSJ + Brown.

Em Tan, *et al* 2017 (32), os autores utilizam um mecanismo de autoatenção (33) para combinar representações geradas por redes neurais artificiais, redes neurais recorrentes e redes neurais convolutivas. A motivação por trás do uso do mecanismo de atenção é contornar duas limitações das RNRs. A primeira delas está relacionada com a compressão de memória Cheng, *et al* 2016 (34). Como toda a memória do contexto está codificada em um vetor de tamanho fixo, isto implica que para que haja representação efetiva de contextos mais longos, maior capacidade de memória faz-se necessária. Em contrapartida, para sentenças menores, esse excesso de capacidade torna-se ocioso. A segunda delas está relacionada com a estrutura interna das sentenças que é hierárquica, RNR não são equipadas para representar estruturas do tipo árvore e o número de não linearidades será sempre relacionada ao número de palavras. A Figura 2.4 reproduz a visualização de um mecanismo de atenção, diferentes seções da sentença ligadas à palavra atual *making*, estão ativadas (coloridas), diferentes *heads* (linhas) atendem à diferentes porções da rede. De acordo com os autores o sistema DeepAtt é capaz de representar dependências globais nas proposições de entrada. Portanto, duas palavras arbitrariamente distantes podem interagir de forma por menores caminhos ( $O(n)$  vs  $O(1)$ ). DeepAtt obteve uma pontuação  $F_1$  86.1% no conjunto de teste oficial WSJ e pontuação  $F_1$  74.80% no conjunto de teste extra oficial Brown para um total combinado de 84,60% da *CoNLL 2005 Shared Task*.

Em He, *et al* 2018 (35), os pesquisadores integram os resultados Lee, *et al.* (36) para a tarefa de Resolução de Coreferência (37, 38). O sistema é capaz de prever conjuntamente predicados e argumentos, para tanto constrói representações para *spans* de argumentos e para predicados. As palavras passam por uma camada de *embeddings* e são alimentadas para uma RNR, as saídas da rede servem de entrada para uma ANN. O sistema obteve um *score*  $F_1$  86,00% no conjunto de teste oficial (WSJ) e *score*  $F_1$  76,1% no conjunto de teste Brown da *CoNLL 2005 Shared Task*.

Em Straubell, *et al* 2018 (39), os pesquisadores retornam à abordagem multitarefas (40). Esta decisão foi tomada após a análise de trabalhos anteriores ((21) entre outros) que indicam que aumentos substanciais em

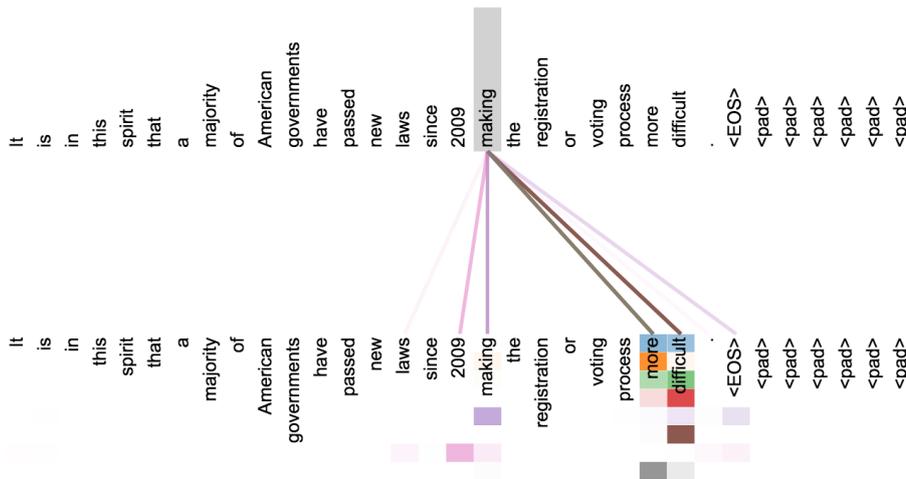


Figura 2.4: Sentença segundo um mecanismo de atenção (33).

acurácia podem ser obtidos, com a reintrodução de atributos sintáticos. O sistema LISA<sup>3</sup>, combina a abordagem multitarefas com os mecanismos de atenção (33) e desempenha as seguintes tarefas: (i) realiza a predição conjunta do predicado e da PoS (ii) *parsing* sintático (iii) atenta aos parentes sintáticos do *token* e (iv) os associa aos devidos papéis semânticos. De forma que camadas inferiores realizam a tarefa de detecção do predicado e PoS e associam os papéis semânticos em paralelo ao invés de em série. LISA obteve um *score*  $F_1$  86,90% no conjunto de teste oficial (WSJ) e *score*  $F_1$  74,66% no conjunto de teste Brown. da *CoNLL 2005 Shared Task* estebelecendo o novo estado da arte.

A Tabela 2.3.1 resume os resultados da pesquisa em Anotação de Papéis Semânticos para o Inglês ordenados, em ordem decrescente, a partir do *score*  $F_1$  no conjunto de teste do WSJ. As colunas Comp. se referem à proporção dos exemplos preditos de forma perfeita. Avaliação no conjunto de teste Brown também foram disponibilizados. A coluna W + B é o *score*  $F_1$  dos dois *datasets* combinados.

	WSJ		Brown		W + B
	$F_1$	Comp.	$F_1$	Comp.	$F_1$
Tan <i>et al</i> , 2017	86,10%	69.0%	74,8%	48,6%	84,6%
Straubell <i>et al</i> , 2018	86,04%	-	76,57%	-	-
He <i>et al</i> , 2018	86,00%	-	76,10%	-	-
He <i>et al</i> , 2017	84,6%	66.5%	73.6%	46.5%	83%
Zhou e Xu, 2015	82,8%	-	69.4%	-	81%
Koomen <i>et al</i> , 2005	79,4%	53,8%	67,8%	32,2%	77,9%
Collobert <i>et al</i> , 2011	75,49%	-	-	-	-

Tabela 2.1: *Leaderboard* para sistema de APS para o inglês.

<sup>3</sup>*Linguistically-informed self-attention*

### 2.3.2

#### Idioma: Português

Em Alva Manchego, 2013 (20), é proposto um modelo baseado em SVM, é composto de 4 subtarefas que conjuntamente possibilitam a anotação de papéis semânticos; (i) Identificação do predicado, identifica qual palavra serve como predicado da proposição. (ii) Poda, isto é a identificação de palavras, através da árvore de constituintes, que não possuem papel semântico na proposição. (iii) Identificação de Argumentos, treina-se um classificador binário para identificar se uma palavra é argumento para o predicado ou não. (iv) Classificação de Argumentos, nessa fase aloca-se as etiquetas aos candidatos da fase anterior. O trabalho faz amplo uso da árvore sintagmática, conforme surgido na literatura (24, 27) entre outros, gerando atributos derivados como caminho de uma determinada palavra até o verbo. O sistema obtém um *score*  $F_1$  79,70% na tarefa de anotação de papéis semânticos,

No mesmo ano Fonseca, 2013 (41), propõe dois modelos de distribuição de palavras, HAL<sup>4</sup> e RI<sup>5</sup> para converter as palavras em vetores. Estes vetores então são alimentados para uma rede neural convolutiva, conforme Collobert e Weston, 2008 (42), esta rede pode ter um ou dois estágios. A rede com um estágio segmenta os argumentos e os classifica. A rede com dois estágios é um sistema multitarefas; (i) A identificação de predicados. (ii) A classificação dos argumentos. O sistema obtém um *score*  $F_1$  57,25%<sup>6</sup>

Em Garrido, 2017 (22), o autor desenvolve um sistema de sub tarefas capaz de anotar textos brutos. O sistema é baseado em CRF e é composto por uma seqüência de subtarefas: (i) POS tagging, (ii) Identificação de Predicados, (iii) Identificação de *Chunkings* Sintáticos, (iv) Identificação de Argumentos, (v) Classificação de Argumentos. A Figura 2.5 mostra a *pipeline* de tarefas. O sistema obtém um *score*  $F_1$  58,00%.

A Tabela 2.3.2 resume os resultados da pesquisa em Anotação de Papéis Semânticos para o Português ordenados, em ordem decrescente, avaliados com o *script* oficial de avaliação da *ConLL 2004 Shared Task*.

	$F_1$
Alva Manchego, 2013	79,70
Garrido, 2017	58,00
Fonseca, 2013	57,25

Tabela 2.2: *Leaderboard* para sistema de APS para o português.

<sup>4</sup>*Hyperspace Analogue to Language*

<sup>5</sup>*Random Indexing*

<sup>6</sup> Utilizando os scripts da *ConLL 2004 Shared Task* conforme (22) na tarefa de anotação de papéis semânticos.

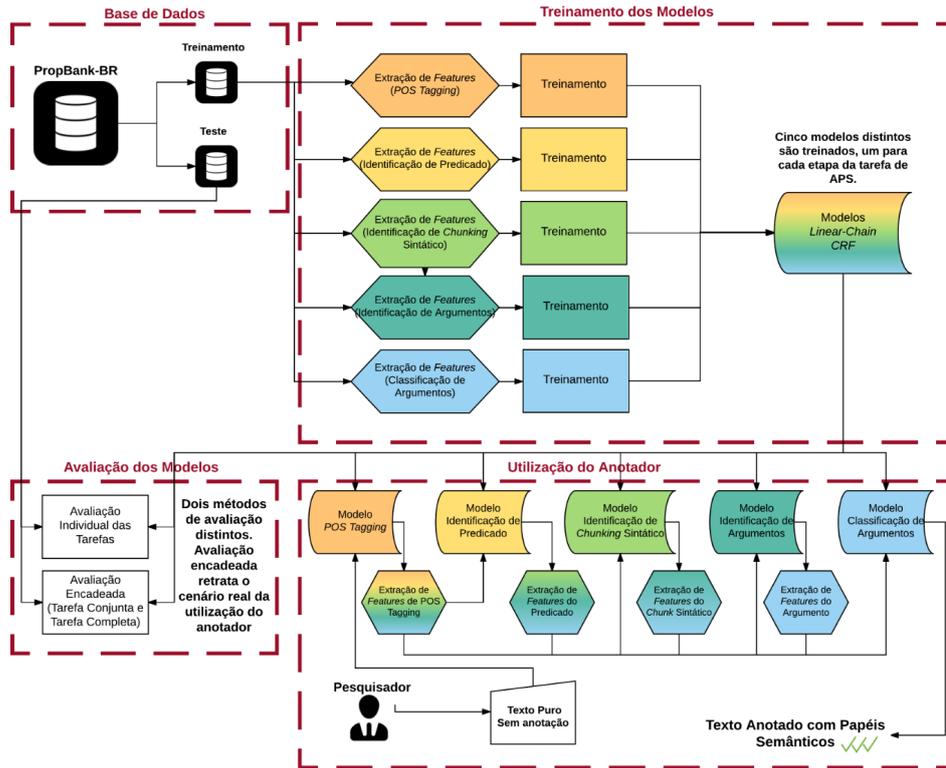


Figura 2.5: Sequência de subtarefas (22).

Diferentemente do que ocorre na literatura dos sistemas em Inglês, o sistema prevalente em termos da métrica  $F_1$  para o Português é amplamente baseado no trabalho de Koomen, *et al* 2005 e portanto faz uso de uma extensiva *pipeline* de tarefas (e.g as saídas de um classificador binário alimenta um classificador de argumentos) Adicionalmente, faz amplo uso da prática de engenharia de atributos, criando atributos sintéticos, a partir de conhecimento prévio linguístico tais como existência ou não de voz passiva. Ou o caminho na árvore sintagmática da palavra até o verbo. A principal razão apontada pelos autores dos trabalhos em português é o baixo número de exemplos. Com poucos exemplos a abordagem da construção automática de representações intermediária pelas camadas de aprendizado profundo perde muito em eficácia.

### 3

## Anotação de Papéis Semânticos

Neste Capítulo restringimos a exposição da teoria lingüística aquela necessária para entender a motivação por trás do desenvolvimento do PropBank. o padrão PropBank é apresentado com exemplos, de forma a esclarecer os desafios enfrentados pelos pesquisadores. Essa dissertação, no entanto, versa sobre o PropBank.Br a variante do PropBank para o português do Brasil. Uma comparação entre os dois modos de anotação é feita dado que os resultados desse trabalho são afetados por eles. A escolha da versão 1.0 é justificada por ser a mesma versão utilizada por outros autores. E finalmente, as estatísticas para o *corpus* são apresentadas.

### 3.1

#### Teoria Lingüística

A descoberta de papéis semânticos foi feita por Panini, um gramático indiano, por volta de 4 A.C para o sânscrito. Mas o interesse dos pesquisadores foi reavivado por Fillmore em 1968. Neste trabalho, o autor defende que as orações são compostas por duas estruturas interrelacionadas: A forma de superfície, ou sintaxe regida pelas funções gramaticais usuais; sujeito, verbo, objeto e advérbio. E a forma profunda – semântica, regida pela gramática de casos. Fillmore identifica 6 desses casos gramaticais e assim como as suas restrições (1):

- Agente (A): O caso em que um ser animado instiga a ação identificado pelo verbo.
- Instrumento (I): O caso em que um força inanimada ou objeto é de forma causal envolvido na ação ou estado identificado pelo verbo.
- Dativo (D): O caso do ser animado sendo afetado pela ação descrita pelo verbo.
- Factivo (F): O caso do objeto ou ser sendo o resultado de alguma ação ou estado identificado pelo verbo, ou compreendido como parte do verbo.
- Locativo (L): O caso que define a localidade ou a orientação espacial do estado ou ação identificados pelo verbo.

- Objetivo (0): O caso mais neutro, o caso de qualquer coisa representável pelo substantivo cujo papel na ação ou estado é identificado pelo verbo é identificável apenas pela interpretação semântica do verbo; provavelmente o conceito deve ser limitado para coisas que são afetadas pela ação ou estado identificado pelo verbo. O termo, não deve ser confundido com a noção objeto direto, nem com o nome do caso de superfície sinônimo ao acusativo.

Cada verbo, está associado à número de casos este relacionamento entre verbos e casos é chamado de *case frames*. Que por sua vez têm as seguintes propriedades:

- Eles ocorrem apenas uma vez na sentença.
- Cada verbo determina quais casos são obrigatórios e quais são opcionais. Casos obrigatórios não podem ser excluídos com o risco de produzir sentenças incorretas gramaticalmente. O verbo **abrir**, requer um AGENTE ou OBJETIVO:

- ✓ [João<sub>A</sub>] abriu [a porta<sub>0</sub>] [com a chave<sub>I</sub>].
- ✓ [A porta<sub>0</sub>] abriu.
- × [João<sub>A</sub>] abriu.
- × [João<sub>A</sub>] abriu [com a chave<sub>I</sub>].

- Hierarquia para a escolha do sujeito:

- [João<sub>A</sub>] abriu [a porta<sub>0</sub>].
- [A chave<sub>I</sub>] abriu [a porta<sub>0</sub>].
- [A porta<sub>0</sub>] abriu.

Estas definições não são exaustivas tendo sido revisadas por (2, 3). Atualmente, não existe um consenso entre pesquisadores com relação ao número de papéis ou ao seu respectivo grau de especificidade. Papéis semânticos têm variado de granularidade de um extremo ligado à uma situações extremamente específicas como por exemplo o verbo prender evoca os papéis de Suspeito, Autoridade e Ofensa em (3). Enquanto Dowty, 1991 (4) defende que os papeis semânticos podem ser descritos por um ou mais atributos de dois Proto-Papéis; o Proto-Agente e o Proto-Paciente conforme a Tabela 3.1. Um AGENTE prototípico teria todas as qualidades de Proto-Agente. Um EXPERIENCIADOR teria as propriedades Proto-Agente (b) e (e) e a propriedade Proto-Paciente (h).

A incerteza gerada pela falta de acordo no conceito linguístico se propaga para o trabalho de anotação automática de papéis semânticos. Papéis menos específicos tais como AGENTE e PACIENTE, são menos controversos e geram

Proto-Agente	Proto-Paciente
a. envolvimento ativo	f. mudança de estado
b. senciência (ou percepção)	g. tema incremental
c. causa mudança de estado	h. afetado de forma causal
d. movimento (relativo à)	i. estacionário (relativo à)
e. existência independente	j. existência dependente

Tabela 3.1: Os Proto-Papéis de Dowty

menos esparsidade nos dados, e.g o AGENTE assume o papel de motorista para o verbo **dirigir** e o papel de orador para o verbo **discursar** (5).

Em Levin, 1993 (6) o foco do estudo dá-se na categorização de verbos em classes distintas. A hipótese levantada por Levin é que verbos que compartilham o mesmo padrão de alternância sintática retêm certa semelhança semântica. Por exemplo, verbos que admitem alternância entre tema (0) e locativo (L) pertencem a classe *spray*<sup>1</sup>. Verbos da classe *spray*, como *load* ou *paint*, envolvem mover alguma coisa para dentro ou sobre uma localidade e ocorre em verbos que encapsulam a idéia de colocação e cobertura. A alternância no entanto expressa diferenças entre completude e parcialidade, com uma variação é interpretada (ii) como enchendo um contêiner ou localidade completamente e outra alternância (i) não tem essa interpretação, conforme Figura 3.1. Enquanto os verbo **acertar** admite alternância entre tema (0) e

PUC-Rio - Certificação Digital Nº 1621788/CA

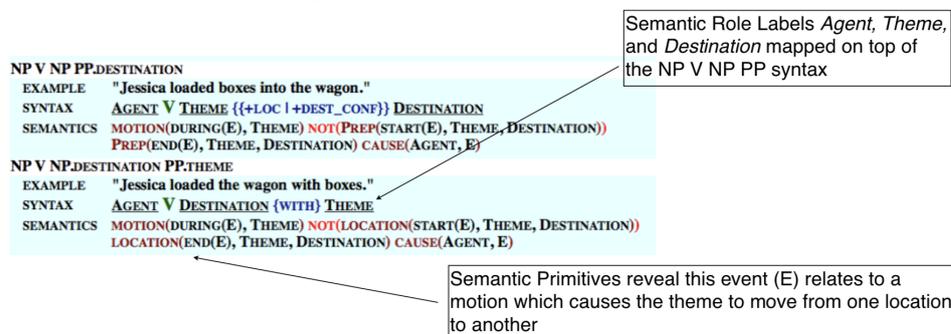


Figura 3.1: Classe de verbo *spray*.

instrumento (I) enquanto **quebrar** não.

- Luís acertou a cerca com o martelo.
- Luís acertou o martelo contra a cerca.
- Luís quebrou o muro com o martelo.
- Luís quebrou o martelo contra o muro.

<sup>1</sup>[http://verbs.colorado.edu/verb-index/VerbNet\\_Guidelines.pdf](http://verbs.colorado.edu/verb-index/VerbNet_Guidelines.pdf)

Verbos são classificados de maneira hierárquica de acordo com a sua assinatura sintática, baseados nesses tipos de alternância. Ainda que as classes de verbo de Levin estejam explicitamente hierarquizadas de acordo com a estrutura de alternância sintática, verbos em classes similares retêm alguma semelhança semântica. O verbo **quebrar** implica mudança de estado e está associado à alguma causa. O verbo **acertar** está associado ao movimento e ao contato enquanto o verbo **tocar** implica a idéia de contato.

Levin, categorizou 3.100 verbos em 47 classes para o inglês. O suficiente para cobrir metade dos verbos do Penn Treebank. Outras limitações incluem; poucos sentidos para cada verbo e verbos ou expressões polissêmicos encontram pouca representatividade. E finalmente, algumas classes de verbo distintas possuem assinaturas sintáticas idênticas (7).

O trabalho de Levin não apenas serviu para a construção do léxico VerbNet mas também estabeleceu a fundação teórica para a criação das *frame files* que compõe o PropBank.

### 3.2 PropBank

PropBank é um *corpus* composto por proposições, que são os predicados (verbos) anotados e seus argumentos (papéis semânticos). Como cada sentença é composta por um ou mais predicados – cada conjunto de atributos pode ter mais que uma saída. Todas as anotações são feitas sobre o as árvores frasais do Penn TreeBank Marcus, et al 1993 (26) no *sub-corpus* de finanças referente ao Wall Street Journal (WSJ). O processo de anotação tem três partes principais: a rotulação de argumentos, a rotulação de modificadores adverbiais e a escolha de um sentido para o predicado. O objetivo de fornecer os argumentos juntamente com os sentidos é manter a consistência entre diferentes realizações sintáticas do mesmo verbo:

- [João<sub>ARG0</sub>] quebrou [a janela<sub>ARG1</sub>].
- [A janela<sub>ARG1</sub>] quebrou.

A rotulação de argumentos consiste na alocação de um argumento numerado A0, A1, ..., A5 para categorizar um papel semântico. Como supramencionado os papéis dependem do predicado e de um sentido. A Figura 3.2 trás os significados dos argumentos para os predicados comprar e vender (43): Uma generalização entre todos os argumentos não é possível, mas usualmente tem-se que os argumentos numerados assumem os papéis semânticos de acordo com a Tabela 3.3.

A rotulação de modificadores adverbiais do tipo AM consiste na escolha de seguimentos que atuam modificando o verbo tais como maneira (MNR),

	Comprar	Vender
A0	Comprador	Vendedor
A1	Coisa comprada	Coisa vendida
A2	Vendedor	Comprador
A3	Preço Pago	Preço Pago
A4	Beneficiário	Beneficiário

Tabela 3.2: Descrição dos argumentos dos verbos comprar e vender

A0	agente
A1	paciente
A2	instrumento, beneficiário ou atributo
A3	ponto de partida, beneficiário ou atributo
A4	ponto de chegada
A5	modificador

Tabela 3.3: Papéis usualmente assumidos pelos argumentos

locativo (LOC) e temporal (TMP) conforme exemplo abaixo (44), enquanto a Tabela 3.4 trás a listagem completa dos argumentos adjuntos do tipo ARG-M:

O sr. Bush o encontrou privadamente, na Casa Branca, na quinta-feira.

- A0: sr. Bush
- A1: o
- AM-MNR: privadamente
- AM-LOC: na Casa Branca
- AM-TMP: na quinta-feira

AM-ADV	propósito geral
AM-CAU	causa
AM-DIR	direção
AM-DIS	discurso
AM-EXT	extensão
AM-LOC	localidade
AM-MNR	maneira
AM-MOD	modo
AM-NEG	negação
AM-PNC	propósito
AM-PRD	predicador
AM-REC	recíproco
AM-TMP	temporal

Tabela 3.4: Papéis adjuntos (45)

Adicionalmente às anotações dos papéis semânticos, e dos modificadores de verbo o padrão estabelece que um sentido deve ser dado aquele predicado naquela proposição. Os *rolesets* consistem na tripla; predicado, sentido e argumentos. Um verbo polissêmico pode ter mais de um *roleset* quando as diferenças em significado são distintas o suficientes à ponto de exigir um conjunto distinto de papel para cada *roleset* (43) segue o exemplo de dois *rolesets* do verbo em inglês *leave*<sup>2</sup>:

*leave.01: move away from*

- ARG0: *entity leaving*
- ARG1: *place left*
- ARG2: *attribute of place left*

Exemplos:

- *Intransitive*: [*John*<sub>ARG0</sub>] left.
- *Transitive*: [*This flight*<sub>ARG0</sub>] leaves [*Phoenix*<sub>ARG1</sub>] [*at midnight*<sub>ARGM-TMP</sub>].
- *Attribute*: [*The move*<sub>ARG0</sub>] left [*companies*<sub>ARG1</sub>] [*as outside bidders*<sub>ARG2</sub>].

*leave.02: to give*

- ARG0: *giver leaver*
- ARG1: *thing given*
- ARG2: *benefactive given to*

Exemplos:

- *Transitive*: [*The buyout*<sub>ARG0</sub>] left [*about \$130 million*<sub>ARG1</sub>] [*in the bank*<sub>ARGM-LOC</sub>].
- *Ditransitive*: [*John*<sub>ARG0</sub>] left [*cookies*<sub>ARG1</sub>] [*Mary*<sub>ARG2</sub>].
- *Double object*: [*John*<sub>ARG0</sub>] left [*Mary*<sub>ARG2</sub>] [*a collection of run-down clocks*<sub>ARG1</sub>].
- *With variant*: [*An ambitious expansion*<sub>ARG0</sub>] has left [*Magma*<sub>ARG2</sub>] [*with excess capacity*<sub>ARG1</sub>].

O exemplo ilustra também as dificuldades na aplicação de *rolesets* entre dois idiomas distintos: No português *leave.01* assume o sentido de ‘partir’ enquanto *leave.02* de ‘deixar’.

A versão do PropBank, utilizada na *ConLL 2005 Shared Task* conta com anotações de papéis semânticos para as seções, do WSJ, 02-21 para treino,

<sup>2</sup> <http://www.lsi.upc.edu/~srlconll/resources/pb-frames.tar.gz>

23 para teste e 24 para validação, adicionalmente três seções do Brown *corpus* foram utilizadas. A Figura 3.2 mostra um exemplo de proposição do PropBank. As colunas se referem à (01) palavras, (02) *Part-of-Speech tag*, (03) Árvore sintagmática completa, (04) Entidades nomeadas, (05) Sentido do verbo de acordo com a *frame file*, (06) Predicados, (07) Argumentos para o verbo *say*, com A0 sendo o falante e A1 é aquilo que foi dito (08) Argumentos para o verbo *brief*, (09) Argumentos para o verbo *leave* com A0 sendo aquele que deixou e AM-DIR foi o destino. Opcionalmente, os atributos de *base chunks* e *clauses* também são fornecidos.

Administration	NN	(S1(S(NP*	*	-	-	(A0*	*	*
officials	NNS	*)	*	-	-	*)	*	*
say	VBP	(VP*	*	1	say	(V*)	*	*
President	NNP	(SBAR(S(NP*	*	-	-	(A1*	(A0*	*
Bush	NNP	*)	(PER*)	-	-	*)	*)	*
was	AUX	(VP*	*	-	-	*	*	*
briefed	VBD	(VP*	*	1	brief	(V*)	*	*
throughout	IN	(PP*	*	-	-	(AM-TMP*	*	*
Friday	NNP	(NP*	*	-	-	*	*	*
afternoon	NN	*	*	-	-	*	*	*
and	CC	*	*	-	-	*	*	*
evening	NN	*)	*	-	-	*)	*	*
,	,	*	*	-	-	*	*	*
even	RB	(SBAR*	*	-	-	*	(AM-TMP*	*
after	IN	*	*	-	-	*	*	*
leaving	VBG	(S(VP*	*	1	leave	*	(V*)	*
for	IN	(PP*	*	-	-	*	(AM-DIR*	*
Camp	NNP	(NP*	*	-	-	*	*	*
David	NNP	*))))))))))	(PER*)	-	-	*)	*)	*
.	.	*)	*	-	-	*	*	*

Figura 3.2: Três proposições do *PropBank*

### 3.3 PropBank.Br

A metodologia de anotação do PropBank.Br seguiu, devido à restrições, um curso diferente ao do PropBank. Onde o primeiro passo foi a construção das *frame files* que descrevem os *frame sets* e a compilação de um guia de anotação, *Anotator's Guidelines*. Desta forma o trabalho pôde ser distribuído, com baixo índice de discordância entre anotadores. Para proposição foi adotado um esquema de anotação duplo-cego, cada desacordo era automaticamente detectado e resolvido por um adjudicador. Em contrapartida, o projeto PropBank.Br (47) adotou o mesmo guia e as *frame files* do Inglês mas para o Português. E na primeira etapa foram anotados apenas os papéis semânticos, sem o sentido do verbo correspondente. A ferramenta de anotação utilizada, tanto para a construção das *frame files* quanto para a anotação do *corpus* foi SALTO (46) desenvolvida para a anotação de um projeto FrameNet para o alemão e ajustada pelos pesquisadores. Em uma segunda fase, 137 *frame files* específicas foram geradas para verbos em português. E camadas de anotações

foram adicionadas posteriormente, como a adição de árvores de dependência e sentido do verbo, dando origem à 4 versões do *corpus* até o momento.

O objetivo dos pesquisadores Magali Duran e Sandra Aluísio (47) ao desenvolver o PropBank.Br é a criação de anotadores automáticos. E como o PropBank uma nova camada de anotação foi adicionada à um *corpus* anotado sintaticamente e revisado manualmente. O *corpus* escolhido foi a porção Brasileira do Bosque, o *sub-corpus* manualmente revisado da Floresta Sintá(c)tica<sup>3</sup> (48) utilizando o *parser* PALAVRAS (49). O Bosque conta com 9368 sentenças, 4213 delas correspondem à variante Brasileira, tendo sido extraídas do jornal Folha de São Paulo em 1994.

O PropBank.Br está disponível em 4 versões distintas. Para manter o caráter comparativo à outros trabalhos, optou-se utilizar a versão 1.0 da base, porém abaixo estão listadas todas as versões (22):

- **1.0**: Mesma utilizada por outras implementações de anotação de papel semântico para o português. Dividido em treino e teste com as mesmas partições utilizadas por outros autores (20, 41, 22). Esta versão apresenta as correções realizadas e descritas por (41) em seu trabalho.
- **1.1**: Versão apresenta algumas correções realizadas, identificação do sentido do verbo em português e anotação de árvores de dependência.
- **2.0**: Originado por uma extração do *corpus* PLN-BR sobre árvores não revisadas usando dois anotadores para cada sentença (anotação duplo-cego) (50).
- **3.0**: Novamente geradas pelo *parser* PALAVRAS, o *corpus* utilizado foi retirado pelo particionamento do Buscapé (17), um *corpus* de opiniões de usuários sobre produtos. As árvores sintáticas da amostra não foram revisadas por humanos.

A Figura 3.3 mostra uma sentença do *corpus* PropBank.Br<sup>4</sup> versão 1.0: As colunas se referem à:

1. **ID** Posição da *FORM*, usualmente tempo, começa em 1 à cada nova proposição.
2. **FORM** forma da palavra. Palavras com hífen, e.g ‘quinta-feira’ ou que formam um sentido semântico, e.g ‘Primeiro Mundo’ são unidas pelo caracter ‘\_’.
3. **LEMMA** *gold-standard* é uma normalização da *FORM* as flexões são retiradas de artigos e verbos.

<sup>3</sup><http://linguateca.pt>

<sup>4</sup><http://143.107.183.175:21380/portlex/index.php/en/projects/propbankbringl>

4. **GPOS** *gold-standard* etiquetas part-of-speech.
5. **MORPH** *gold-standard* atributos morfológicos da *FORM*
6. **SENTENCE** *gold-standard* marcadores da sentença.
7. **CLAUSE** *gold-standard* tipo de oração no caso FCL a oração está na forma finita.
8. **CTREE** *gold-standard* árvore sintagmática completa.
9. **PRED** predicados semânticos da proposição.
10. **ARG** argumentos relativos ao predicado *afirmar*.
11. **ARG** argumentos relativos ao predicado *ler*.

Uma descrição completa do conteúdo de cada uma dessas colunas está além do escopo dessa dissertação. Os leitores interessados podem se referir à dissertações (20, 67) ou ao sítio do manual lingüístico da Floresta Sintá(c)tica<sup>5</sup>:

1 <<	-	PU	-	(S(S*	(FCL(FCL*	(FCL(FCL*	-	(A1*	*
2 É	ser	V-FIN	PR 3S IND	*	*	(VP*)	-	*	*
3 uma	um	ART	F S	*	*	(NP*	-	*	*
4 coisa	coisa	N	F S	*	*	*	-	*	*
5 de	de	PRP	-	*	*	(PP*	-	*	*
6 o	o	ART	M S	*	*	(NP*	-	*	*
7 Primeiro_Mundo	Primeiro_Mundo	PROP	M S	*	*	*))	-	*	*
8 >>	-	PU	-	*)	*)	*)	-	*)	*
9 ,	-	PU	-	*	*	*	-	*	*
10 afirmou	afirmar	V-FIN	PS 3S IND	*	*	(VP*)	afirmar	(V*)	*
11 o	o	ART	M S	*	*	(NP*	-	(A0*	*
12 levantador	levantador	N	M S	*	*	*	-	*	*
13 Maurício	Maurício	PROP	M S	*	*	(NP*)	-	*)	*
14 (	-	PU	-	*	*	*	-	*	*
15 leia	ler	V-FIN	PR 1/3S SUBJ	(S*	(FCL*	(FCL(VP*)	ler	*	(V*)
16 matéria	matéria	N	F S	*	*	(NP*)	-	*	(A1*)
17 a	a	PRP	-	*	*	(PP*	-	*	(AM-LOC*
18 o	o	ART	M S	*	*	(NP*	-	*	*
19 lado	lado	N	M S	*)	*)	*))	-	*	*)
20 )	-	PU	-	*	*	*	-	*	*
21 .	-	PU	-	*)	*)	*)	-	*	*

Figura 3.3: Exemplo do PropBank.Br

Adicionalmente, a mesma partição utilizada para essa dissertação, é a mesma utilizada para os autores (20, 22, 41)

<sup>5</sup><https://www.linguateca.pt/Floresta/BibliaFlorestal/>

	Treino	Validação	Teste
Sentenças	3.011	151	141
Proposições	5.299	235	239
Verbos Distintos	989	148	165
Argumentos	12.520	553	541
A0	2.820	121	125
A1	4.794	214	213
A2	1.027	45	38
A3	106	7	2
A4	66	8	1
A5	0	0	0
AM-ADV	338	13	20
AM-CAU	150	6	1
AM-DIR	12	1	2
AM-DIS	271	13	11
AM-EXT	78	3	1
AM-LOC	722	29	28
AM-MED	0	0	0
AM-MNR	377	17	18
AM-NEG	302	16	19
AM-PNC	159	7	5
AM-PRD	179	10	7
AM-REC	60	0	5
AM-TMP	1.059	43	45

Tabela 3.5: PropBank.Br v1.0 partições.

## 4 Aprendendo Seqüências

O sistema de anotação automática de papéis semânticos aqui proposto é inspirado em Zhou e Xu (9), e composto por três camadas; (i) modelos para representação de palavras, (ii) uma rede neural recorrente do tipo LSTM com múltiplas camadas, (iii) uma camada de predição estruturada. Os sistemas anteriores para a tarefa para o idioma português são baseados em explorar os seguintes aspectos: atributos sintáticos alimentados para um separador linear SVM (20); tirar proveito da estrutura hierárquica da linguagem, ao fazer uso de redes convolutivas (41); explorar as múltiplas subtarefas através de um *pipeline* (22). Aqui, a motivação é elaborar uma arquitetura que dispensa o uso de atributos linguísticos, ou a implementação de uma seqüência de subtarefas.

### 4.1 Representação de Palavras

A representação das palavras é um aspecto fundamental em qualquer modelo de aprendizado de máquina no domínio da PLN, pois ela tem forte impacto na qualidade do aprendizado. Especificamente em AAPS, as primeiras abordagens (24, 51, 45) contavam com a representação do tipo *one-hot encoding*. Neste caso, dado o vocabulário  $\mathbf{V}$  com todas as palavras do *corpus*, onde  $V = |\mathbf{V}|$  é o número de palavras,  $\mathbf{X} \in \mathbb{R}^{V \times V}$ ,  $\mathbf{X}_{\bullet k}$  é o vetor coluna representando a  $k$ -ésima palavra de  $\mathbf{V}$ , e  $x_{ik}$  é o  $i$ -ésimo elemento desse vetor, temos então

$$x_{ik} = \begin{cases} 1, & \text{se } i = k \\ 0, & \text{caso contrário} \end{cases} \quad (4-1)$$

Um exemplo era então composto de *tokens* assim representados eram concatenados com atributos. Esses exemplos por sua vez, eram representados por pontos no espaço hiperdimensional, e alimentados à um separador linear (8, 27, 20). No entanto, esse tipo de representação é proibitivo para redes neurais, dado que o tempo de treino e inferência está associada à complexidade em tempo de execução por sua vez que está associada ao espaço de parâmetros. A dimensionalidade dos parâmetros está diretamente ligada ao tamanho do vocabulário  $V$ , e finalmente o tamanho do vocabulário pode ser arbitrariamente grande.

Apenas para o idioma inglês é estimada a existência de 13 milhões de *tokens*, por exemplo.

Uma outra limitação da representação é que elas carecem de qualquer semântica. Bengio, *et al* (52) propõem o conceito de *word embeddings* fundamentados na hipótese distributiva – palavras que coocorrem dividem o mesmo sentido semântico. Todos os modelos de representação de palavras, nessa seção expostos, têm como objetivo estimar a probabilidade de ocorrência, de uma palavra ou conjunto de palavras dado o contexto em que aparece. Uma forma de converter representações vetoriais esparsas, em representações densas, com sentido semântico é através de um modelo *word2vec*. Tomamos um caso simples, no qual uma rede neural tem como entrada a palavra atual, e como saída a palavra seguinte.

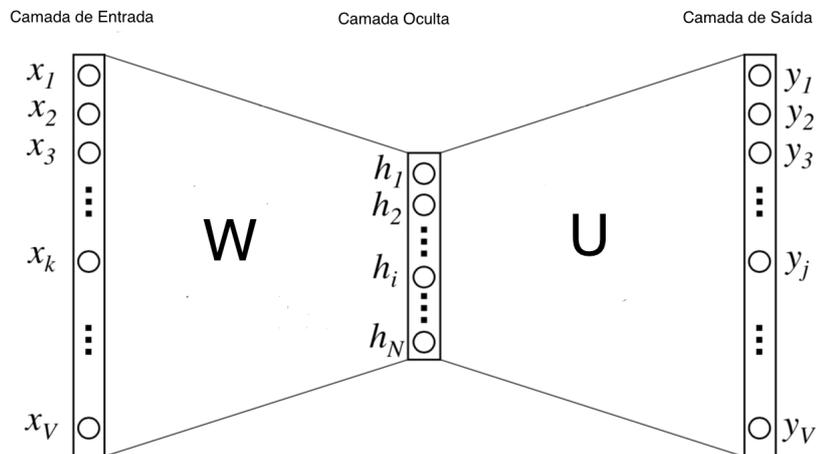


Figura 4.1: *word2vec* (53)

A Figura 4.1 ilustra uma rede do tipo *feed forward* com uma camada oculta, de tamanho  $D \ll V$  e função de ativação linear. Os pesos das camadas de entrada e saída, são dados pelas matrizes  $\mathbf{W} \in \mathbb{R}^{V \times D}$  e  $\mathbf{U} \in \mathbb{R}^{D \times V}$  respectivamente. Neste modelo o contexto, é dado pela próxima palavra, e.g na frase ‘Quando os ventos sopram’ um par válido de treino é (‘ventos’, ‘sopram’), associados aos vetores,  $\mathbf{x}_k \in \mathbf{X}$  e  $\mathbf{y}_j \in \mathbf{X}$ , cabe a rede estimar  $\mathbb{P}(\text{‘sopram’} | \text{‘ventos’})$ , conforme Equações 4-2

$$\begin{aligned}
 \mathbf{h} &= \mathbf{W}^T \mathbf{x}_k = \mathbf{w}_k \\
 \mathbf{u} &= \mathbf{U}^T \mathbf{w}_k \\
 \tilde{y} &= \text{softmax}(\mathbf{u}) = \frac{\exp\{\mathbf{u}\}}{\sum_V \exp\{\mathbf{u}\}}
 \end{aligned}
 \tag{4-2}$$

### 4.1.1 Word2Vec

O modelo *word2vec* foi introduzido por Mikolov, *et al*, 2013 (54) e trata-se de uma flexibilização do contexto  $C$  em torno da entrada ou saída, em relação a rede da Figura 4.1.

#### 4.1.1.1 CBOW: Sacola de palavras contínua

CBOW é uma generalização do modelo representado na Figura 4.1. Onde, ao invés do contexto ser definido, como o *token* antecedente, o contexto tem tamanho fixo  $C$ , e o alvo é o *token* intermediário. Cabe à camada oculta, o papel de construir um resumo das representações. A Figura 4.2 mostra a arquitetura CBOW.

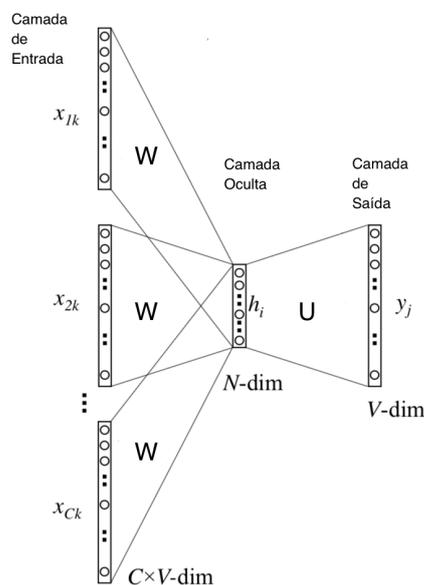


Figura 4.2: CBOW: sacola contínua de palavras (53)

### 4.1.1.2 Skipgram

*Skipgram* pode ser entendido como o modelo recíproco ao CBOW. Onde, dado um token  $\mathbf{x}_k$  a rede deve produzir um contexto de tamanho fixo  $C$ . A Figura 4.3 mostra a arquitetura da rede *Skipgram*.

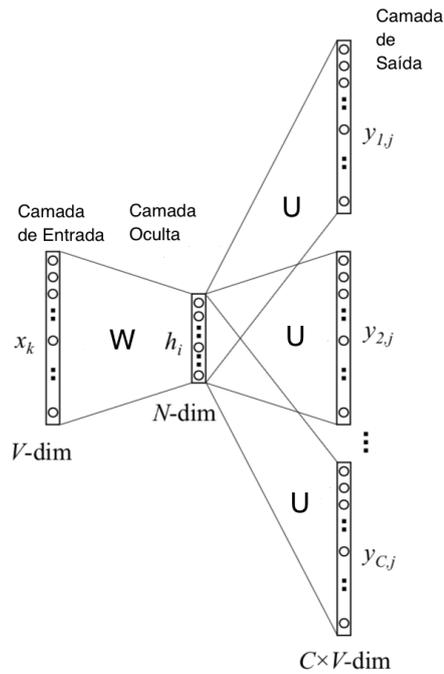


Figura 4.3: *Skipgram* (53)

### 4.1.2 Wang2Vec

Em Ling, *et al* (68), são propostas duas modificações simples para os modelos *word2vec*: CBOW, os vetores coluna representando os *tokens* no contexto, são concatenados. *Skipgram*, é utilizada uma matriz de pesos diferente para cada posição do contexto de saída.

### 4.1.3 GloVe

GloVe é um método proposto por Pennington, *et al* (55), e obteve resultados estado de arte para tarefas sintáticas e semânticas em tarefas de analogia. O método consiste em uma matriz de coocorrência  $\mathbf{M}$  que é construída olhando-se cada *token* em relação ao seu contexto. Cada elemento  $\mathbf{M}_{ij}$  na matriz representa a probabilidade da palavra  $i$  ser próxima à palavra

$j$ . Na matriz  $M$ , as linhas, ou vetores, são geradas aleatoriamente, e treinadas obedecendo a Equação 4-3 (56).

$$\mathbb{P}(\mathbf{w}_i, \mathbf{w}_j) = \mathbf{w}_i \cdot \mathbf{w}_j + b_i + b_j = \log(M_{ij}) \quad (4-3)$$

## 4.2 Redes Neurais Recorrentes

As redes neurais recorrentes são redes especializadas em modelar seqüências que são representadas por uma coleção de vetores cada qual na forma  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(\tau)}$  onde  $\mathbf{x}^{(t)}$  representa um vetor em uma posição  $t$  arbitrária e  $\tau$  o seu comprimento máximo. Neste trabalho o conjunto de desenvolvimento, é composto por proposições. Por sua vez cada proposição é formada por *tokens* e cada token é enriquecido com atributos. Os *tokens* são representados por vetores conforme a Seção 4.1, e que são concatenados aos seus respectivos atributos formando  $\mathbf{x}^{(t)}$ .

O grafo de computação é um recurso visual para ilustrar dependências entre variáveis de entrada, saída e operações e estão formalmente definidos no Apêndice A ou (57). O processo de *forward propagation* de rede neural recorrente pode ser representada pelo grafo de computação na Figura 4.4, (a) versão compacta do processo de *Feed forward* (b) versão desdobrada no tempo. Os parâmetros  $\mathbf{U}, \mathbf{V}, \mathbf{W}$  são compartilhados entre iterações: (i)  $\mathbf{U}$  pesos do entre nós de entrada e oculto (ii)  $\mathbf{W}$  pesos entre nós ocultos (iii)  $\mathbf{V}$  pesos entre nó oculto e observável. As Equações 4-4 definem o relacionamento entre variáveis. O estado  $\mathbf{h}$  representa um resumo das informação até o momento  $t$ . Os interceptos  $\mathbf{b}, \mathbf{c}$  foram omitidos sem perda de generalidade.

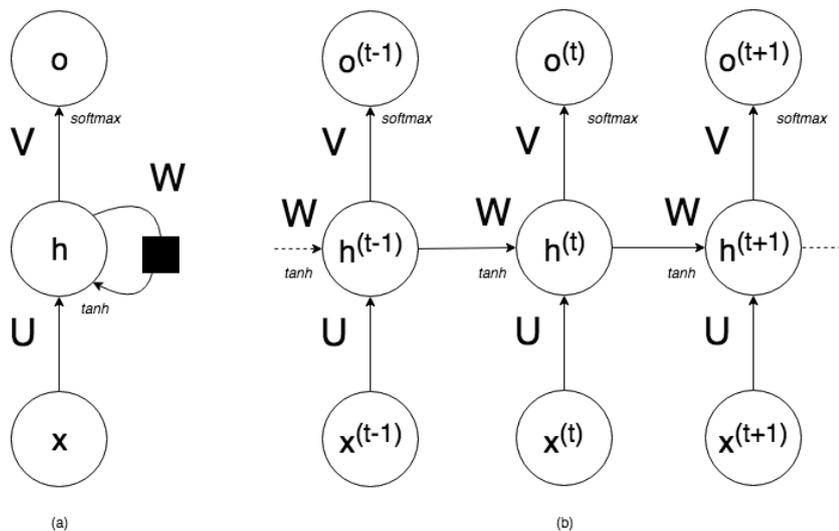


Figura 4.4: O grafo de computação de uma RNR

$$\mathbf{h}^{(t)} = \tanh(\mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)}), \tag{4-4}$$

$$\mathbf{o}^{(t)} = \text{softmax}(\mathbf{V}\mathbf{h}^{(t)})$$

### 4.2.1 Redes Neurais Recorrentes Bidirecionais

As RNRs Bidirecionais como o próprio nome indica, combinam RNRs que se movem do início ao fim da seqüência, i.e para frente no tempo, e RNRs que se movem do fim ao início da seqüência, i.e de trás para frente no tempo. Em alguns casos como reconhecimento de voz e de escrita, modelos se beneficiam em receber a informação dos vetores sucedentes da seqüência para decidir a saída atual (57). Especificamente, na tarefa de anotação de papéis semânticos, o sistema pode tirar proveito que a etiqueta A0 já foi realizada no passado, e que a etiqueta A2 já foi realizada no futuro, para gerar uma pontuação favorável para a etiqueta A1.

As Bi RNRs contam com duas estruturas de recorrência siamesas, a Figura 4.5 (a) demonstra duas relações de recorrência uma causal (sentido horário) e a outra não causal (sentido antihorário), cada qual gera uma pontuação normalizada  $\mathbf{o}, \mathbf{z}$  que é concatenada no vetor  $\mathbf{y}$  (b) grafo de computação desdobrado. Para cada instante  $t$ , o nó oculto  $\mathbf{h}^{(t)}$  armazena um resumo da informação passada e o nó oculto  $\mathbf{g}^{(t)}$  armazena um resumo da informação futura. Como na Figura 4.4 os parâmetros  $\mathbf{U}, \mathbf{W}, \mathbf{V}$  são compartilhados, apenas entre a RNR que avança no tempo. Enquanto  $\mathbf{K}, \mathbf{I}, \mathbf{J}$  são compartilhados, apenas entre a RNR que retroage pelo tempo.

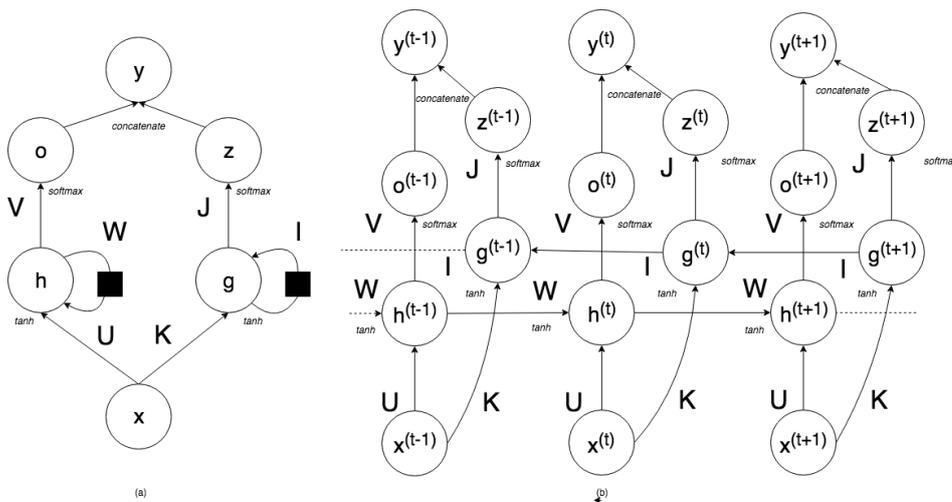


Figura 4.5: O grafo de computação de uma Bi RNR

## 4.2.2

**O problema do gradiente minguento explosivo**

Um desafio para treinar redes profundas é o problema do gradiente minguento ou explosivo (PGME) que está associado ao algoritmo de retro-propagação (*backpropagation*) (58, 59). Cabe ao algoritmo computar os valores de ajustes nos pesos da rede, através de aplicações da regra da cadeia, de forma a diminuir uma função de perda. Como o produto de vários fatores menores que um, pode rapidamente convergir para zero, o ajuste de pesos nas camadas mais profundas serão mitigados, resultando em uma não diminuição da função de perda. Redes neurais recorrentes, sofrem do problema de forma ainda mais aguda, uma vez que o aprendizado se dá através de uma variação do algoritmo de retro-propagação, o chamado algoritmo de retro propagação através do tempo (BPTT: *backpropagation through time*). Onde o ajuste de pesos ocorre com respeito ao grafo de computação desenrolado, através do tempo, e.g Figura 4.5 (b). Como as sentenças podem ser arbitrariamente longas, assim também o ajuste de pesos deve ocorrer em um número arbitrário de passos. Para ilustrar o problema basta tomar a relação mais simples da relação de recorrência na Equação 4-5 (57), com  $\mathbf{W} \in \mathbb{R}^{d \times d}$  e  $\mathbf{h} \in \mathbb{R}^d$

$$\mathbf{h}^{(t)} = \mathbf{W}^T \mathbf{h}^{(t-1)} \quad (4-5)$$

A Equação 4-5 pode ser reescrita em função do estado de partida  $\mathbf{h}^{(0)}$  e pela potência dos parâmetros  $\mathbf{W}$ :

$$\mathbf{h}^{(t)} = (\mathbf{W}^{(t)})^T \mathbf{h}^{(0)} \quad (4-6)$$

Se a matriz  $\mathbf{W}$  é positiva definida então ela admite fatoraçoão ortonormal com  $\mathbf{\Lambda} \in \mathbb{R}^{d \times d}$  matriz diagonal com os autovalores e  $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}$  e  $\mathbf{I}$  matriz identidade, então  $\mathbf{W}$ , pode ser reescrita como:

$$\mathbf{W} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \quad (4-7)$$

E finalmente temos que:

$$\mathbf{h}^{(t)} = \mathbf{Q}^T \mathbf{\Lambda}^{(t)} \mathbf{Q} \mathbf{h}^{(0)} \quad (4-8)$$

$\mathbf{\Lambda}^{(t)}$  explode para autovalores maiores que um e mingua para autovalores menores que zero a demonstraçãõ se encerra.

### 4.2.3 Long Short Term Memory

Uma variação da redes neurais recorrentes inspiradas em unidades de memória RAM, foi proposta por Hochreiter e Schmidhuber (60). A Figura 4.6 ilustra um grafo de computação das LSTM (*Long Short Term Memory*). Três portas foram criadas para facilitar o fluxo do gradiente no grafo de computação. São as portas; (i) *forget*  $f$  que quando ativada remove o conteúdo do estado interno (ii) *input*  $g$  controla a contribuição da entrada no estado interno (iii) *external*  $q$  controla o que é propagado para a saída. Adicionalmente, a rede conta com um estado de saída  $h$  e um estado interno  $s$ . As variáveis ( nós ) anônimos no grafo foram criados para facilitar a visualização. O sistema de Equações 4-9 descreve a interação entre portas e parâmetros que ocorre dentro da unidade. Já o sistema de Equações 4-10 descreve a interação entre as diferentes portas e estados da LSTM.

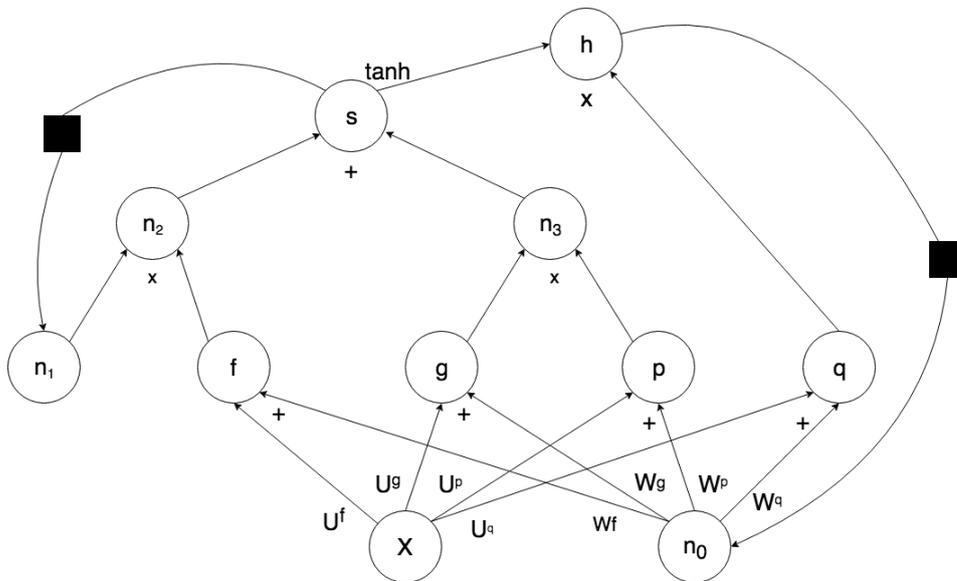


Figura 4.6: O grafo de computação de uma LSTM

$$\begin{aligned}
\mathbf{f} &= \sigma(\mathbf{U}^f \mathbf{x} + \mathbf{W}^f \mathbf{n}_0) \\
\mathbf{g} &= \sigma(\mathbf{U}^g \mathbf{x} + \mathbf{W}^g \mathbf{n}_0) \\
\mathbf{p} &= \sigma(\mathbf{U}^p \mathbf{x} + \mathbf{W}^p \mathbf{n}_0) \\
\mathbf{q} &= \sigma(\mathbf{U}^q \mathbf{x} + \mathbf{W}^q \mathbf{n}_0) \\
\mathbf{n}_2 &= \mathbf{f} \times \mathbf{n}_1 \\
\mathbf{n}_3 &= \mathbf{g} \times \mathbf{p} \\
\mathbf{s} &= \mathbf{n}_2 + \mathbf{n}_3 \\
\mathbf{h} &= \tanh(\mathbf{s}) \times \mathbf{q}
\end{aligned} \tag{4-9}$$

$$\begin{aligned}
\mathbf{f}^{(t)} &= \sigma(\mathbf{U}^f \mathbf{x}^{(t)} + \mathbf{W}^f \mathbf{h}^{(t-1)}) \\
\mathbf{g}^{(t)} &= \sigma(\mathbf{U}^g \mathbf{x} + \mathbf{W}^g \mathbf{h}^{(t-1)}) \\
\mathbf{p}^{(t)} &= \sigma(\mathbf{U}^p \mathbf{x} + \mathbf{W}^p \mathbf{h}^{(t-1)}) \\
\mathbf{q}^{(t)} &= \sigma(\mathbf{U}^q \mathbf{x} + \mathbf{W}^q \mathbf{h}^{(t-1)}) \\
\mathbf{s}^{(t)} &= \mathbf{f}^{(t)} \times \mathbf{s}^{(t-1)} + \mathbf{g}^{(t)} \times \mathbf{p}^{(t-1)} \\
\mathbf{h}^{(t)} &= \tanh(\mathbf{s}^{(t)}) \times \mathbf{q}^{(t)}
\end{aligned} \tag{4-10}$$

As portas efetivamente criam caminhos por onde o computo do gradiente pode fluir(60), mitigando o PGME. Na prática, as LSTMs foram adotadas pelos pesquisadores e o seu uso se tornou quase um sinônimo para redes neurais recorrentes.

### 4.3 Predição Estruturada

Em aprendizado de máquina usualmente o interesse está em modelar o comportamento de uma variável de saída  $\mathbf{y}$ , seja ela discreta ou contínua, em função de um conjunto de atributos  $\mathbf{x}$ . Essa arranjo implicitamente reflete uma importante premissa – que o valor da variável  $\mathbf{y}$  depende apenas de  $\mathbf{x}$ , e que não existe interdependência entre observações passadas e atuais de  $\mathbf{y}$ . É a prerrogativa da predição estruturada gerar saídas mais ricas que variáveis discretas ou contínuas, tais como variáveis discretas ordenáveis (61), intervalos (62), árvores ou grafos (63).

Em AAPS, sob o padrão *PropBank*, a tarefa consiste em simultaneamente segmentar e classificar intervalos. Como a saída deve representar intervalo, o comportamento dos papéis semânticos é interdependente. Um outro exemplo são as classes dado que dois intervalos, na mesma proposição, não podem

assumir a etiqueta **A1** por exemplo. E pesquisadores fizeram largamente o uso de restrições lingüísticas em seus sistemas (8) (27) (21). Essa seção expõe os dois modos de predição estruturadas pertinentes à essa dissertação.

### 4.3.1

#### Conditional Random Fields

Modelos Gráficos é uma família de distribuições de probabilidade que são fatoradas por um grafo associado. A idéia geral é representar uma distribuição sobre um conjunto grande de variáveis aleatórias por um produto de funções locais que dependem apenas de um pequeno subconjunto de variáveis (64).

**Definição 4.1 *Conditional Random Field*** *Um distribuição de probabilidade  $p(\mathbf{x}, \mathbf{y})$  pode ser representado por variáveis aleatórias  $V = X \cup Y$  onde  $X$  é um conjunto de variáveis de entradas observadas e  $Y$  é um conjunto de variáveis de saída.  $\mathbf{x}$  é uma realização de  $X$  em particular  $\mathbf{x}_A$  é uma realização que pertence ao subconjunto  $A \subset V$ , e analogamente para  $Y$ .*

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_A \Psi_A(\mathbf{x}_A, \mathbf{y}_A) \quad (4-11)$$

A constante  $Z$  é um fator de normalização definido por:

$$Z = \sum_{\mathbf{x}, \mathbf{y}} \prod_A \Psi_A(\mathbf{x}_A, \mathbf{y}_A) \quad (4-12)$$

CRFs podem representar cadeias de Markov conforme a equação 4-13. As cadeias de Markov fazem duas premissas de independência sobre o conjunto de dados  $X$  e  $Y$ ; (i) cada estado dependente apenas de seu predecessor, i.e  $p(y_t|y_{t-1}, y_{t-2}, \dots, y_1, y_0) = p(y_t|y_{t-1})$  (ii) a probabilidade de emissão  $p(x_t|y_t)$  depende apenas do seu estado atual  $y_t$ .

$$p(\mathbf{x}, \mathbf{y}) = \prod_{t=1}^T p(y_t|y_{t-1})p(x_t|y_t) \quad (4-13)$$

A equação 4-13 pode ser escrita na forma 4-11:

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp \left\{ \sum_t \sum_{i,j} \lambda_{i,j} \mathbb{I}_{\{y_t=i\}\{y_{t-1}=j\}} + \sum_t \sum_{i \in S} \sum_{o \in O} \mu_{i,j} \mathbb{I}_{\{x_t=o\}\{y_t=i\}} \right\} \quad (4-14)$$

com

$$\begin{aligned}\theta &= \{\lambda_{i,j}, \mu_{o,i}\} \\ \lambda_{i,j} &= \log p(y = i | y' = j) \\ \mathbb{I}_{x=x'} &= \begin{cases} 1, & x = x' \\ 0, & \text{cc} \end{cases}\end{aligned}$$

**Definição 4.2 *linear-chain conditional random field*** Seja  $Y, X$  vetores aleatórios,  $\Lambda = \{\lambda_k\} \in R^K$  um vetor de parâmetro, e  $\{f_k(y, y', \mathbf{x}_t)\}_{k=1}^K$  seja um conjunto de funções reais. Então um campo aleatório condicional do tipo cadeia linear *linear-chain conditional random field* tem distribuição condicional  $p(\mathbf{y}|\mathbf{x})$  que toma a forma:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left\{ \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\} \quad (4-15)$$

Onde  $Z(\mathbf{x})$  é uma função de normalização, específica para o conjunto de dados:

$$Z(\mathbf{x}) = \sum_y \exp \left\{ \sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\} \quad (4-16)$$

Pode ser mostrado que, as equações 4-15 e 4-16 acomodam as premissas da cadeia de Markov fazendo (64):

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp \left\{ \sum_t \sum_{i,j} \lambda_{i,j} \mathbb{I}_{\{y_t=i\}\{y_{t-1}=j\}} + \sum_t \sum_{i \in S} \sum_{o \in O} \mu_{o,i} \mathbb{I}_{\{x_t=o\}\{y_t=i\}} \right\} \quad (4-17)$$

$$Z(\mathbf{x}) = \sum_y \exp \left\{ \sum_t \sum_{i,j} \lambda_{i,j} \mathbb{I}_{\{y_t=i\}\{y_{t-1}=j\}} + \sum_t \sum_{i \in S} \sum_{o \in O} \mu_{o,i} \mathbb{I}_{\{x_t=o\}\{y_t=i\}} \right\} \quad (4-18)$$

A função densidade de probabilidade conjunta  $p(\mathbf{x}, \mathbf{y})$  na Equação 4-14 fatora uma HMM, enquanto a função de desidade de probabilidade marginal nas 4-17 e 4-18 fatora uma *linear-chain conditional random field*. A família de parâmetros  $\theta = \{\lambda_{i,j}, \mu_{o,i}\}$  pode ser estimada por gradiente descendente, programação dinâmica ou otimizando por máxima verossimilhança.

### 4.3.2 Perceptron Estruturado

Um *perceptron* (65) é uma representação matemática para um neurônio humano. Trata-se de uma função determinada por uma entrada  $\mathbf{x} \in \mathbb{R}^n$  um vetor de pesos  $\mathbf{w} \in \mathbb{R}^n$ , um escalar (viés)  $\mathbf{b} \in \mathbb{R}$  e uma função de disparo  $g(\nu) \in \{-1, 1\}$ . Tanto os pesos como o viés podem ser estimados pelo algoritmo do *perceptron*, à essa estimativa de parâmetros é dado o nome de aprendizado.

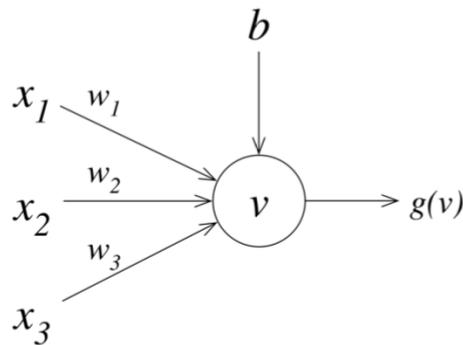


Figura 4.7: O *perceptron* (62)

$$\nu = \sum_{i=1}^n w_i x_i + b \quad (4-19)$$

Sem perda de generalidade tomando-se  $w_0 = b$  e  $x_0 = 1$  pode-se escrever:

$$\nu = \sum_{i=0}^n w_i x_i = \mathbf{w} \cdot \mathbf{x} \quad (4-20)$$

E a função de disparo  $g$  pode ser definida como:

$$g(\nu) = \begin{cases} -1 & \text{se } \nu < 0 \\ +1 & \text{se } \nu \geq 0 \end{cases}$$

As redes neurais modernas ainda são constituídas por camadas de *perceptrons*, com variações na função de disparo. Embora sejam treinadas através do método do gradiente descendente.

Em 2002, Collins (66) estende o lema de convergência do *perceptron*, generalizando a função de disparo  $g(\nu)$  para uma função  $\Phi(\mathbf{x}, \mathbf{y})$ . A Figura 1 mostra o algoritmo de treino, i.e ajustes de pesos, para o *perceptron* estruturado. Para cada par  $(\mathbf{x}_i, \mathbf{y}_i)$  no conjunto de treino  $D$ , a previsão é feita escolhendo-se o  $\hat{\mathbf{y}}_i$  que maximize o produto  $\omega \cdot \Phi(\mathbf{x}_i, \hat{\mathbf{y}}_i)$ . O lema de convergência do *perceptron* estruturado garante que se  $\Phi(\mathbf{x}, \mathbf{y})$  separa o pontos no espaço

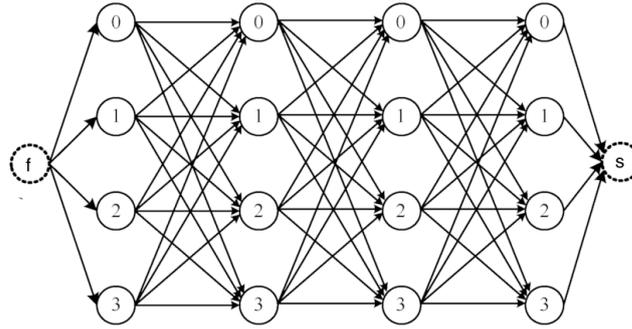


Figura 4.8: Treliça

então  $\omega$  converge. Modelar um problema utilizando o *perceptron* estruturado significa definir  $h(\mathbf{x})$  na Equação 4-21:

$$\hat{\mathbf{y}} = h(\mathbf{x}) = \operatorname{argmax}_{z \in \mathcal{Y}(\mathbf{x})} \{\mathbf{w} \cdot \Phi(\mathbf{x}, z)\} \quad (4-21)$$

**Data:**  $\{(\mathbf{x}, \mathbf{y})\}$   
**Result:**  $\mathbf{w}$   
**while** *no convergence* **do**  
     $\mathbf{w} \leftarrow 0$  **foreach**  $(\mathbf{x}_i, \mathbf{y}_i)$  **do**  
         $\hat{\mathbf{y}} \leftarrow \operatorname{argmax}_{z \in \mathcal{Y}(\mathbf{x})} \{\mathbf{w} \cdot \Phi(\mathbf{x}, z)\}$   
         $\mathbf{w} \leftarrow \mathbf{w} + \Phi((\mathbf{x}_i, \mathbf{y}_i)) - \Phi((\mathbf{x}_i, \hat{\mathbf{y}}))$   
    **end**  
**end**

**Algorithm 1:** Treino do Perceptron Estruturado

Em 2018, Milidiú e Rocha (63), propõe uma aplicação do *perceptron* estruturado para modelar seqüências. Reescrevendo Equação 4-21 como:

$$\hat{\mathbf{y}} = h(\mathbf{x}) = h_1(h_0(\mathbf{x}; \Theta)) \quad (4-22)$$

Onde  $h_0(\mathbf{x}; \Theta)$  são os **custos latentes** induzidos pela estrutura nesse caso um grafo direcionado acíclico, em camadas chamadol de treliça Figura 4.8 e  $\Theta$  o seu conjunto associado de parâmetros.  $h_1$  **camada estruturada**, responsável por encontrar  $\hat{\mathbf{y}}$  no espaço de soluções viável  $\mathcal{Y}(\mathbf{x})$  através da resolução de um problema de otimização combinatória.

A estrutura é dada pela treliça, que é construída a partir dos exemplos (proposições)  $\{(\mathbf{x}, \mathbf{y})\}$  no *corpus*  $\mathcal{D}$  tem comprimento de  $\tau$  *tokens*. Dois nós especiais são introduzidos; **f** fonte e **s** sorvedouro, pontos de partida e chegada respectivamente na seqüência de etiquetas  $\mathcal{Y}$ . Na Figura 4.8, os nós  $\{0, 1, 2, 3\}$

correspondem a representação de  $k=4$  classes possíveis para  $\mathcal{Y}$  e a proposição tem comprimento  $\tau = 4$ , dado pelo número de camadas do grafo.

Com a estrutura definida, o próximo passo para definir  $h_0(\mathbf{x}; \Theta)$  é escolher os átomos – a parte da estrutura que receberá atributos. Nesse caso serão as arestas (63), cada aresta pode ser denominada como o par  $(u, v)$ . O mapa ou descrição para cada aresta é dado por  $\phi(\mathbf{x}, t, u, v)$ , com  $\mathbf{x}$  atributos para cada  $t$ ,  $t$  a posição dentro da proposição, e  $u$  e  $v$  duas classes em  $\mathbf{Y}$  então o mapa dado pela primeira aresta é  $\phi(\mathbf{x}, 0, f, y_0)$  o da última é  $\phi(\mathbf{x}, \tau, y_\tau, s)$  com  $y_0$  e  $y_\tau$  em  $\mathbf{y}$ . E em geral os mapas induzidos pela estrutura são dados por  $\phi(\mathbf{x}, t, u, v)$  com  $\mathbf{x} \in \mathcal{X}$ , e  $\mathcal{X}$  conjunto admissível de todas as combinações possíveis de atributos de tamanho  $\tau$ ,  $t = 0 \dots \tau$  e  $u, v \in \mathcal{Y}(\mathbf{x})$ ,  $\mathcal{Y}(\mathbf{x})$  conjunto de todas as seqüências possíveis de classe de tamanho  $\tau$ , ou caminhos na treliça.

A Figura 4.9 apresenta um exemplo, como entrada temos três atributos: (i)  $A_1$  conjunto de palavras, (ii)  $A_2$  função indicadora se o verbo da proposição foi encontrado, (iii)  $A_3$  verbo. A entrada  $\mathbf{x}_t$  é obtida concatenando-se os elementos para cada  $t = 0 \dots 4$ , com  $\tau = 4$ . A treliça é construída tomando-se o conjunto de todas as seqüências de tamanho  $\tau = 4$  de  $k = 4$  classes  $\mathbf{Y}^4$ . O conjunto de todos os caminhos  $\mathcal{Y}$  dado por Equação 4-23

$$\mathcal{Y} = \{(y_0, y_1, y_2, y_3, y_4) | y_0 = \mathbf{f} \wedge y_4 = \mathbf{s} \wedge y_i \in \mathbf{Y}^4, i = 1 \dots 3\} \quad (4-23)$$

Os mapas de custo deveriam dar-se para cada um dos tempos pelas Figuras 4.10- 4.14

Já a camada estruturada  $h_1$  é definida pela assinatura  $\Phi(\mathbf{x}, \mathbf{y})$ , dada pela soma dos mapas para cada aresta. E o  $argmax$  é dado pela resolução de um problema de maior caminho. Se  $\mathbf{z} \equiv (z_1, \dots, z_\tau) \in \mathcal{Y}$ , representa a seqüência de nós relacionada a saída  $\mathbf{y}$  então:

$$\mathbf{z} = \cup_{t=0, \dots, \tau} \{(t, z_t, z_{t+1})\} \quad (4-24)$$

Com  $z_0 = f$ ,  $z_{\tau+1} = s$ , então o custo correspondente pode ser escrito como:

$$\Phi(\mathbf{x}, \mathbf{z}) = \sum_{t=0}^{\tau} \phi(\mathbf{x}, t, z_t, z_{t+1}) \quad (4-25)$$

A partir da função assinatura na Equação 4-25 podemos gerar os custos  $c(\mathbf{x}, t, z_t, z_{t+1})$  (63)

$$w \cdot \Phi(\mathbf{x}, \mathbf{z}) = \sum_{t=0}^{\tau} c(\mathbf{x}, t, z_t, z_{t+1}) \quad (4-26)$$

$A_1$  (algumas, pessoas, constroem, moinhos)

$A_2$  (0, 0, 1, 1)

$A_3$  (constroem, constroem, constroem, constroem)

$\gamma^4$  (B-A0, I-A0, V, B-A1)

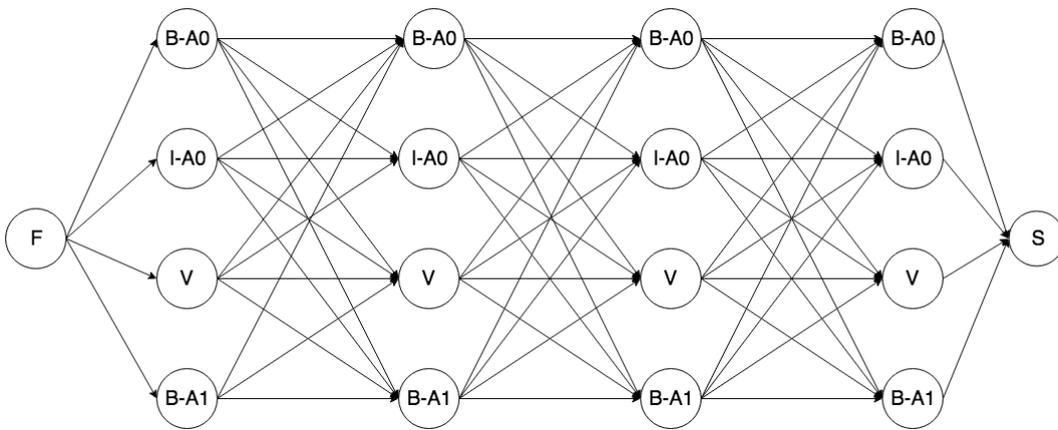
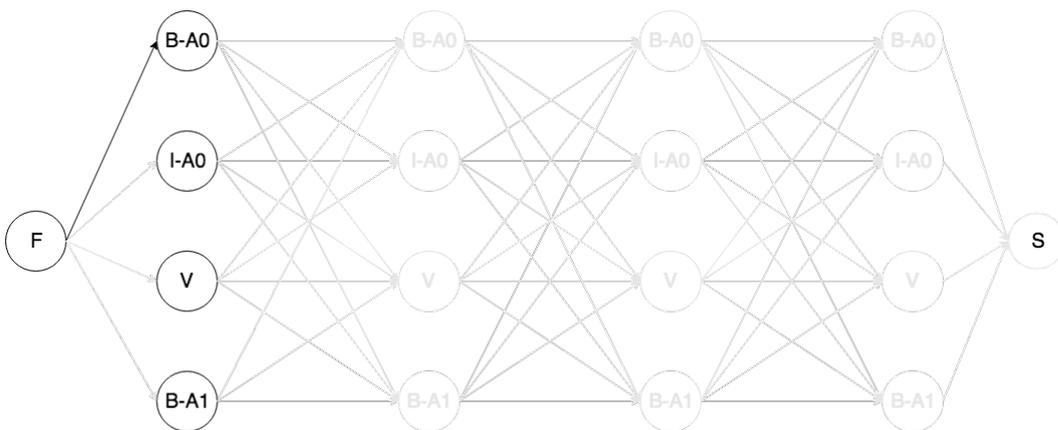


Figura 4.9: Treliça



$\phi((\text{algumas}, 0, \text{constroem}), 0, f, B-A0)$

Figura 4.10: Mapa de custo  $t = 0$

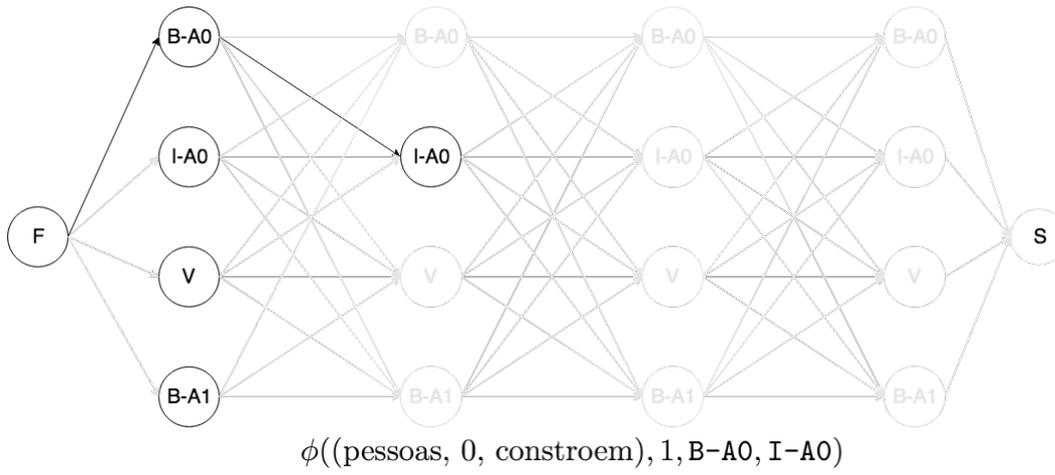


Figura 4.11: Mapa de custo  $t = 1$

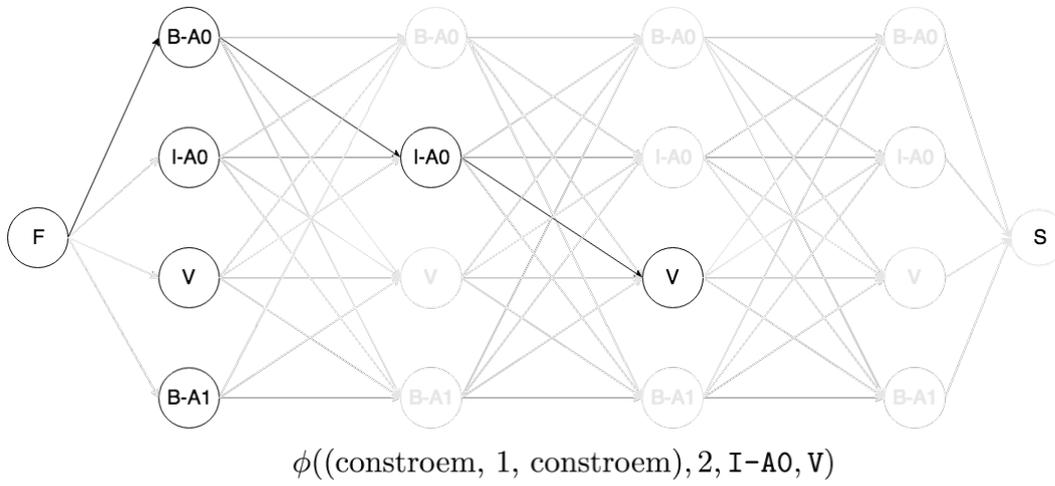


Figura 4.12: Mapa de custo  $t = 2$

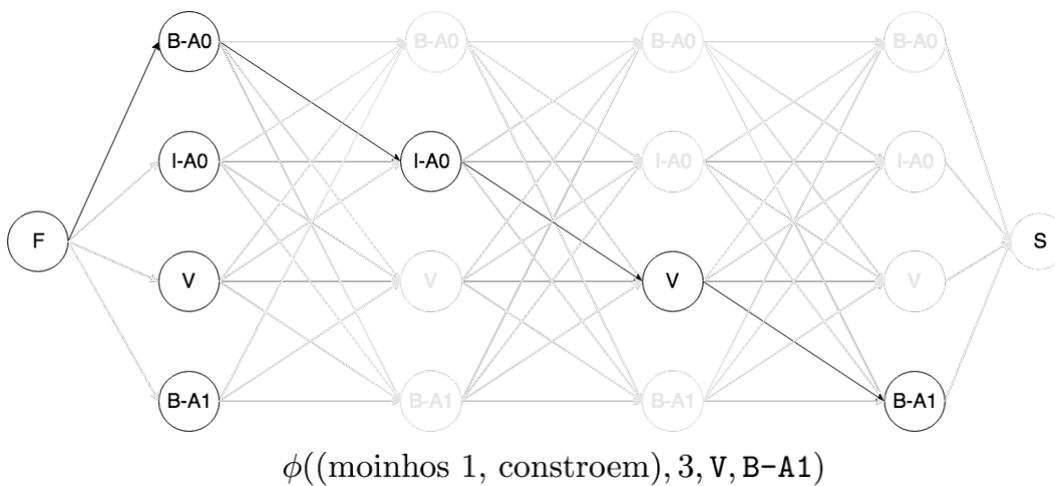


Figura 4.13: Mapa de custo  $t = 2$

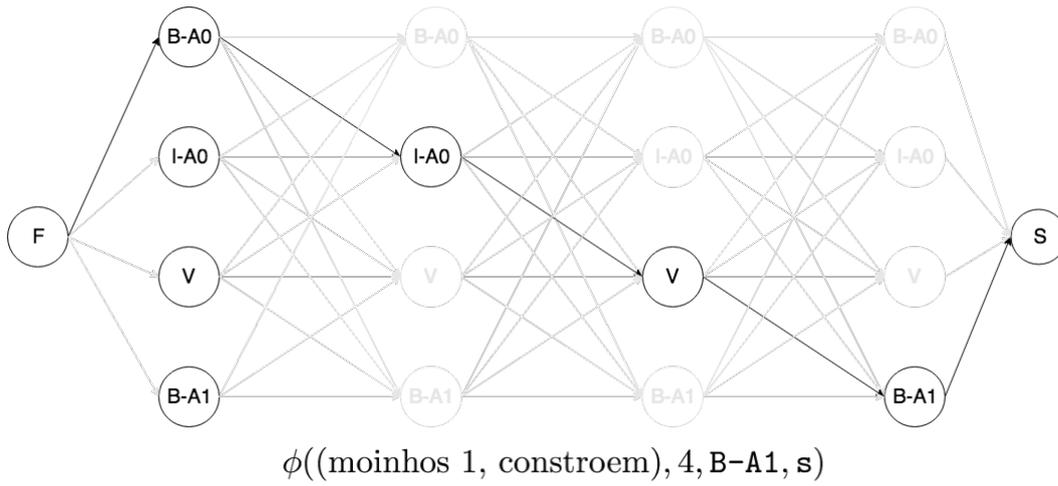


Figura 4.14: Mapa de custo  $t = 4$

Combinando as Equações 4-21, 4-25 e 4-26, teremos:

$$\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{z} \in \mathcal{Y}(\mathbf{x})} \sum_{t=0}^{\tau} c(\mathbf{x}, t, z_t, z_{t+1})$$

Onde *argmax* é resolvido através da resolução de um problema de maior caminho. Concluindo a demonstração.

## 5 Metodologia

O objetivo dessa dissertação é aplicar para o português do Brasil, a arquitetura de aprendizado profundo através de redes neurais recorrentes que estabeleceu-se como estado da arte, em 2015, para o idioma do inglês. Neste Capítulo finalmente especificamos o sistema de aprendizado profundo de papéis semânticos.

### 5.1 Pré processamento

O pré processamento têm três partes; a seleção de atributos padrão ouro, a conversão de *tokens* em *embeddings* e a criação de atributos artificiais.

#### 5.1.1 Seleção de atributos padrão ouro

Um das características dessa dissertação é fazer um uso parcimonioso de atributos linguísticos. De fato o único utilizado são as etiquetas *Part-of-speech* padrão ouro. A escolha pela sua utilização deu-se pela crença de que podem completar o sentido dado pela semântica dos *embeddings* e por outro lado podem ser gerados por uma tarefa com alta acurácia. Os atributos padrão ouro utilizados são:

- FORM: São as palavras do *corpus* tokenizadas.
- GPOS: São *Part-of-speech* padrão ouro. A Tabela 5.1 resume os valores possíveis.
- PRED: É o verbo que atua como predicado na proposição, esse predicado ainda será processado conforme Seção 5.1.3.

O resultado desta etapa está ilustrado no primeiro quadro da Figura 5.1. Nessa dissertação tomamos a sub tarefa de reconhecimento de predicado alvo da Seção 2.2.5 como resolvida – a mesma premissa da competição da CoNLL. Como a sentença da Figura 3.3 tem duas proposições ela gera dois exemplos no conjunto de dados pós processado.

GPOS	Descrição
substantivos	n
substantivos/adjetivos	n-adj
adjetivos	adj
nomes próprios	prop
advérbios	adv
verbos finitos	v-fin
verbos gerúndios	v-ger
verbos particípios	v-pp
verbos infinitivos	v-inf
artigos	art
pronomes determinativos	pron-det
pronomes relativos	pron-rel
pronomes pessoais	pron-pess
preposições	prep
interjeições	intj
conjunções subordinativas	conj-c
conjunções coordenativas	conj-s
prefixos	ec
pontuações	pu

Tabela 5.1: *Part-of-Speech*

**1ª ETAPA**

EXEMPLO 1				EXEMPLO 2			
FORM	GPOS	PRED	ARG	FORM	GPOS	PRED	ARG
<<	PU	-	(A1*	<<	PU	-	*
É	V-FIN	-	*	É	V-FIN	-	*
uma	ART	-	*	uma	ART	-	*
coisa	N	-	*	coisa	N	-	*
de	PRP	-	*	de	PRP	-	*
o	ART	-	*	o	ART	-	*
Primeiro_Mt	PROP	-	*	Primeiro_Mt	PROP	-	*
>>	PU	-	*)	>>	PU	-	*
,	PU	-	*	,	PU	-	*
afirmou	V-FIN	afirmar	(V*)	afirmou	V-FIN	-	*
o	ART	-	(AO*	o	ART	-	*
levantador	N	-	*	levantador	N	-	*
Maurício	PROP	-	*)	Maurício	PROP	-	*
(	PU	-	*	(	PU	-	*
leia	V-FIN	-	*	leia	V-FIN	ler	(V*)
matéria	N	-	*	matéria	N	-	(A1*)
a	PRP	-	*	a	PRP	-	(AM-LOC*
o	ART	-	*	o	ART	-	*
lado	N	-	*	lado	N	-	*)
)	PU	-	*	)	PU	-	*
.	PU	-	*	.	PU	-	*

Figura 5.1: Pré processamento 1ª etapa.

### 5.1.2

#### Transformação de *tokens*

Nem todas as palavras pertencentes ao *corpus* estão presentes nos *embeddings* utilizados e produzidos por (56). A conversão de palavras em *tokens* geralmente recai sobre uma das regras abaixo:

- conversão em letra minúscula : todos os tokens estão em letra minúscula.
- horários, tempos e ordinais: foram convertidos para o caracter ‘0’

- palavras compostas: e.g Primeiro\_Mundo foram convertidas para minúscula e o valor do vetor de *embeddings* equivale à média dos vetores de *embeddings* que a compõe, no caso ‘primeiro’ e ‘mundo’.
- palavras não encontradas: são substituídas por ‘unk’.

Os exemplos após essa etapa encontram-se na Figura 5.2 no segundo quadro.

2a ETAPA							
EXEMPLO 1				EXEMPLO 2			
FORM	GPOS	PRED	ARG	FORM	GPOS	PRED	ARG
<<	PU	-	(A1*	<<	PU	-	*
é	V-FIN	-	*	é	V-FIN	-	*
uma	ART	-	*	uma	ART	-	*
coisa	N	-	*	coisa	N	-	*
de	PRP	-	*	de	PRP	-	*
o	ART	-	*	o	ART	-	*
primeiro_mu	PROP	-	*	primeiro_mu	PROP	-	*
>>	PU	-	*)	>>	PU	-	*
,	PU	-	*	,	PU	-	*
afirmou	V-FIN	afirmar	(V*)	afirmou	V-FIN	-	*
o	ART	-	(A0*	o	ART	-	*
levantador	N	-	*	levantador	N	-	*
unk	PROP	-	*)	unk	PROP	-	*
(	PU	-	*	(	PU	-	*
leia	V-FIN	-	*	leia	V-FIN	ler	(V*)
matéria	N	-	*	matéria	N	-	(A1*)
a	PRP	-	*	a	PRP	-	(AM-LOC*
o	ART	-	*	o	ART	-	*
lado	N	-	*	lado	N	-	*)
)	PU	-	*	)	PU	-	*
.	PU	-	*	.	PU	-	*

Figura 5.2: Pré processamento 2<sup>a</sup> etapa.

### 5.1.3

#### Extração de atributos

Nesta etapa seguimos o exemplo de Zhou e Xu (9) onde dois atributos foram criados para auxiliar as redes neurais recorrentes à controlar posição do *token* com relação ao predicado e o contexto em torno do predicado. Os atributos são:

1. *MARKER* trata-se de uma função indicadora marcando se o predicado da proposição foi visto ou não.
2. *CTX\_P* uma janela em torno do predicado, contendo os *tokens* mais à direita e mais à esquerda que são repetidos para cada passo no tempo.

Adicionalmente os *tokens* a saída é representada no formato BIO. O resultado após a etapa de extração de atributos pode ser na Figura 5.3.

		3ª ETAPA						
		FORM	GPOS	MARKER	CTX_P-1	CTX-0	CTX_P+1	BIO
EXEMPLO 1	<<	PU	0	,		afirmar	o	B-A1
	é	V-FIN	0	,		afirmar	o	I-A1
	uma	ART	0	,		afirmar	o	I-A1
	coisa	N	0	,		afirmar	o	I-A1
	de	PRP	0	,		afirmar	o	I-A1
	o	ART	0	,		afirmar	o	I-A1
	primeiro_mu	PROP	0	,		afirmar	o	I-A1
	>>	PU	0	,		afirmar	o	I-A1
	,	PU	0	,		afirmar	o	O
	afirmou	V-FIN	1	,		afirmar	o	B-V
	o	ART	1	,		afirmar	o	B-A0
	levantador	N	1	,		afirmar	o	I-A0
	unk	PROP	1	,		afirmar	o	I-A0
	(	PU	1	,		afirmar	o	O
	leia	V-FIN	1	,		afirmar	o	O
	matéria	N	1	,		afirmar	o	O
	a	PRP	1	,		afirmar	o	O
	o	ART	1	,		afirmar	o	O
	lado	N	1	,		afirmar	o	O
	)	PU	1	,		afirmar	o	O
.	PU	1	,		afirmar	o	O	
EXEMPLO 2	<<	PU	0	(		ler	matéria	O
	é	V-FIN	0	(		ler	matéria	O
	uma	ART	0	(		ler	matéria	O
	coisa	N	0	(		ler	matéria	O
	de	PRP	0	(		ler	matéria	O
	o	ART	0	(		ler	matéria	O
	primeiro_mu	PROP	0	(		ler	matéria	O
	>>	PU	0	(		ler	matéria	O
	,	PU	0	(		ler	matéria	O
	afirmou	V-FIN	0	(		ler	matéria	O
	o	ART	0	(		ler	matéria	O
	levantador	N	0	(		ler	matéria	O
	unk	PROP	0	(		ler	matéria	O
	(	PU	0	(		ler	matéria	O
	leia	V-FIN	1	(		ler	matéria	B-V
	matéria	N	1	(		ler	matéria	B-A1
	a	PRP	1	(		ler	matéria	B-AM-LOC
	o	ART	1	(		ler	matéria	I-AM-LOC
	lado	N	1	(		ler	matéria	I-AM-LOC
	)	PU	1	(		ler	matéria	O
.	PU	1	(		ler	matéria	O	

Figura 5.3: Pré processamento 3ª etapa.

## 5.2 Redes Neurais Recorrentes

A motivação do uso de redes neurais recorrentes é a geração automática de atributos e que elas explicitamente acomodam exemplos, proposições, de tamanho variável. Neste trabalho testamos camadas do tipo BiLSTM (Seção 4.2.1) e DBLSTM (9). Diferente do Capítulo 4 os diagramas nessa Seção não fazem uso do grafo de computação.

### 5.2.1 BiLSTM

O diagrama da Figura 5.4 mostra duas camadas de uma rede BiLSTM.  $X_t$  representa um vetor de atributos de entrada correspondente à t-ésima linha

da Figura 5.3 por exemplo.  $S_t$  representa um *score* ou uma pontuação gerada automaticamente pela rede referente ao momento  $t$ . As unidades  $h_t^{(c)}$  movem-se para frente no tempo enquanto as unidades  $g_t^{(c)}$  movem-se no sentido inverso, em ambas  $c$  indica a camada. Os círculos com  $\bullet$  representam a operação de concatenação de vetores.

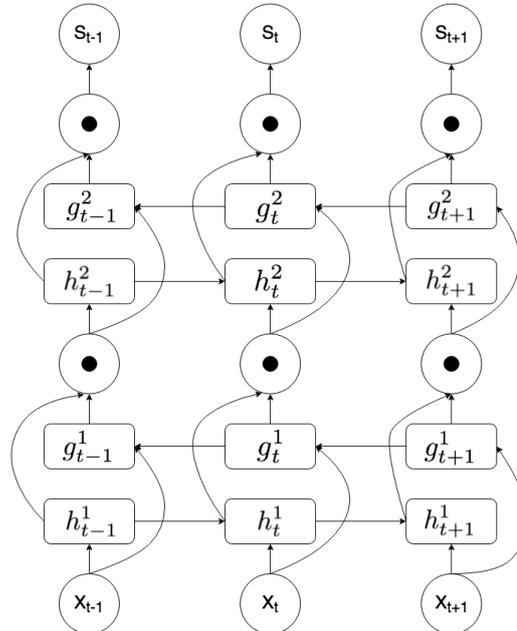


Figura 5.4: BiLSTM Diagrama

### 5.2.2 DBLSTM

As DBLSTM foram introduzidas por Zhou e Xu (9) e segundo os autores essa estrutura é essencial para a resolução da tarefa para o idioma do inglês. O diagrama da Figura 5.5 mostra duas camadas de uma rede DBLSTM, seguindo a mesma nomenclatura da Figura 5.4. DBLSTM significa *Dual Bidirectional LSTM* e é caracterizada pelo padrão entrelaçado entre camadas. Cada camada têm profundidade 2 e os autores defendem que tal arranjo é necessário para obter o alto desempenho da tarefa.

### 5.3 Predição Estruturada

Enquanto o relacionamento intertemporal de *tokens* e atributos está contemplada pelas redes neurais recorrentes e que o contexto dos *tokens* influencia as classes candidatas. Uma outra pergunta pertinente é: ‘Como as escolha das classes anteriores impacta na escolha atual?’. Para a tarefa de anotação automática de papéis pesquisadores fazem largo uso de camadas

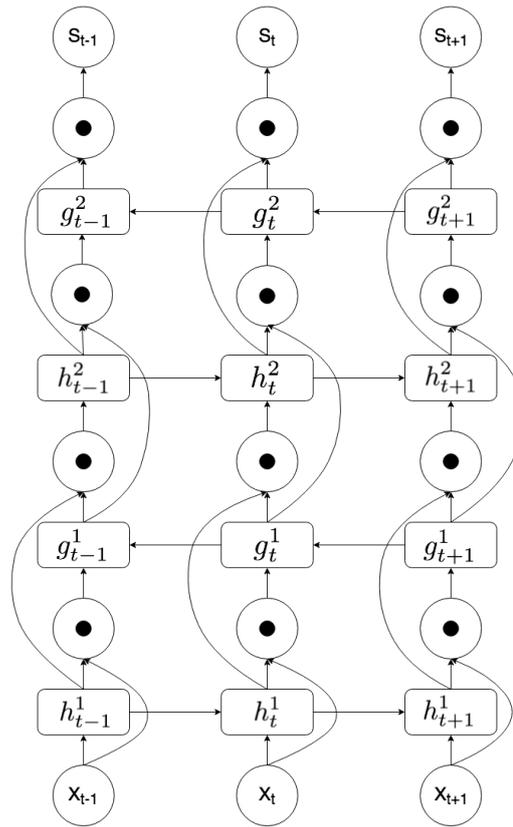


Figura 5.5: DBLSTM Diagrama

de predição estruturada (9) (21). A abordagem tem a dupla vantagem de permitir modelar o impacto de escolhas de classes passadas na escolha atual e inclusive reforçar restrições como proibir dois seguimentos com etiquetas **A1**. Nessa dissertação testamos CRF e a treliça conforme apresentado no Capítulo 4 Seção 4.3.

## 6

# Resultados Experimentais

Neste Capítulo comparamos os resultados do sistema ao de outros autores para o PropBank.Br. O código foi escrito na linguagem de programação Python utilizando o *framework* TensorFlow. Os testes foram executados em um Intel(R) Core(TM) i7-5960X CPU @ 3.00GHz 64 GB RAM o *learning rate* foi mantido fixo em  $5 \cdot 10^{-4}$  por simplicidade. A motivação foram experimentos preliminares indicaram que os resultados são pouco sensíveis a variações deste hiper parâmetro. O otimizador escolhido foi o *AdamOptimizer* (69). Os experimentos foram treinados por Gradiente descendente estocástico, mantendo o tamanho da amostra fixo em 250. E todos os modelos de palavras foram treinados através da variante CBOW. O *script* utilizado para avaliação dos resultados é o *script* da *CoNLL Shared Task 2004* – o mesmo utilizado por (22) escrito na linguagem de programação Pearl.

### 6.1

#### Etapa 1

Na primeira fase do teste mantemos o modelo de *embeddings* fixo no *wang2vec* de tamanho 50. O nosso objetivo é testar: Qual é o tipo de rede neural recorrente DBLSTM ou BiLSTM que desempenha melhor? Quais são os tamanhos e a profundidade, mais apropriados para as camadas escondidas? Quais das duas camadas de predição estruturada CRF ou SPN modela melhor a estrutura das saídas?

Os resultados da primeira fase, encontram-se na Tabela 6.1 ordenados em ordem decrescente por pontuação  $F_1$ . O padrão mais evidente está na coluna P.E (predição estruturada) – a camada CRF superou o *perceptron* estruturado. Entre os 10 melhores modelos nenhum deles contava com SPN. SPN é um modelo que permite tem uma maior expressividade: Na CRF a probabilidade de transição é fixa enquanto no SPN a transição é modelada através de atributos. Aparentemente o espaço de busca não contém um regularidade suficiente para que os parâmetros possam se ajustar. Discernir entre as camadas do tipo DBLSTM e BiLSTM, já não é tão simples, já que entre os 10 melhores se encontram igualmente representados com 5 registros cada. E ainda os modelos que tendem a ter melhor desempenho tiveram profundidade 2 e tamanhos de

256 e 512. Estes resultados se chocam com os publicados por (9) onde foi informado que a estrutura de DBLSTM é crucial para o desempenho e que o modelo final tinha um tamanho de camada escondida de 32 e profundidade 4. Estas camadas ocultas tem portanto tamanho muito superior. A configuração vencedora nessa primeira fase foi BiLSTM com profundidade 2 e camadas ocultas de tamanho 256 cada, e cuja previsão estruturada deu-se por CRF. Obtendo uma pontuação  $F_1$  66,25% no conjunto de validação. Concluímos que a BiLSTM de profundidade 2 e tamanho 252 e modelo de predição estruturada por CRF tiveram o melhor desempenho.

	$F_1$		modelo de palavras		LSTM		P.E
	validação	treino	nome	dimensão	camada	tamanho	modelo
1	66,25	88,3	wang2vec	50	bilstm	256x256	crf
2	63,55	89,33	wang2vec	50	dblstm	512x512	crf
3	63,52	92,29	wang2vec	50	bilstm	128x128	crf
4	63,3	87,98	wang2vec	50	dblstm	128	crf
5	62,75	92,99	wang2vec	50	bilstm	512x512	crf
6	62,28	84,7	wang2vec	50	dblstm	256	crf
7	62,24	96,11	wang2vec	50	bilstm	512	crf
8	61,88	82,43	wang2vec	50	bilstm	256	crf
9	61,61	92,7	wang2vec	50	dblstm	256x256	crf
10	60,93	87,31	wang2vec	50	dblstm	512x512	crf
11	60,89	88,84	wang2vec	50	dblstm	512	spn
12	60,84	80,23	wang2vec	50	dblstm	512	crf
13	60,72	73,94	wang2vec	50	dblstm	256x256	crf
14	60,56	82,85	wang2vec	50	dblstm	512	crf
15	60,52	88,91	wang2vec	50	bilstm	128	crf
16	60,3	93,2	wang2vec	50	bilstm	512x512	spn
17	59,74	80,42	wang2vec	50	dblstm	256	crf
18	59,41	84,6	wang2vec	50	dblstm	256	spn
19	59,4	84,77	wang2vec	50	dblstm	64	crf
20	59,06	72,36	wang2vec	50	dblstm	128x128	crf
21	59,01	88,16	wang2vec	50	dblstm	128x128	crf
22	58,98	88,12	wang2vec	50	bilstm	256x256	spn
23	58,34	85,12	wang2vec	50	dblstm	128	spn
24	58,34	92,16	wang2vec	50	dblstm	256x256	crf
25	58,14	83,22	wang2vec	50	dblstm	512x512	spn
26	57,45	85,06	wang2vec	50	dblstm	256x256	spn
27	57,27	90,62	wang2vec	50	dblstm	64x64	crf
28	56,86	83,19	wang2vec	50	bilstm	128x128	spn
29	56,12	81,28	wang2vec	50	dblstm	128x128	spn
30	55,94	81,49	wang2vec	50	dblstm	128	spn
31	55,91	85,79	wang2vec	50	dblstm	512x512	spn
32	55,82	75,4	wang2vec	50	bilstm	512	spn
33	55,57	81,28	wang2vec	50	dblstm	32x32	crf
34	55,54	82,82	wang2vec	50	bilstm	256	spn
35	55,46	84,31	wang2vec	50	dblstm	256	spn
36	55,37	81,51	wang2vec	50	dblstm	256	spn
37	54,71	83,59	wang2vec	50	dblstm	512	crf
38	54,71	72,11	wang2vec	50	dblstm	32	spn
39	53,92	72,84	wang2vec	50	dblstm	64x64	spn
40	52,66	77,1	wang2vec	50	dblstm	128x128	spn
41	52,66	70,35	wang2vec	50	bilstm	128	spn
42	52,47	73,67	wang2vec	50	dblstm	64	spn
43	51,14	74,02	wang2vec	50	dblstm	128	spn
44	49,09	73,29	wang2vec	50	dblstm	32x32	spn
45	47,54	71,47	wang2vec	50	dblstm	32	spn

Tabela 6.1: Resultados da experimentais 1.

## 6.2

### Etapa 2

Na segunda fase do teste testamos o modelo de *embeddings* entre *word2vec* e 3 tamanhos diferentes na representação no caso do *wang2vec*. O tipo de camada de LSTMs foi fixa na BiLSTM por parcimônia: Ela apresenta melhor desempenho com um número de parâmetros reduzido. No entanto variamos o tamanho e profundidade em busca de uma melhor configuração. Foram testados tamanhos de camadas entre 64 e 400, enquanto a máxima profundidade testada foi de 4. O melhor experimento foi incluído nessa fase para fins de comparação.

Os resultados da segunda fase, encontram-se na Tabela 6.2. Nota-se que ao aumentar a profundidade da rede ou a representação vetorial dos atributos não se consegue superar a configuração anterior. Uma razão apontada por (20, 41, 22), é o baixo número de exemplos: O PropBank.Br tem 5.773 proposições e 13.594 argumentos contra 99.265 proposições e 262.281 argumentos do PropBank (WSJ) (45). A versão americana é portanto cerca de 17 à 19 vezes maior. O fundamento por trás da geração automática de atributos é que existe uma repetição de exemplos suficientemente grande no conjunto de treinamento de tal forma que os exemplos do conjunto de teste estão bem representados. E o tamanho do *corpus* parece agir como um fator limitante. Vale lembrar que o PropBank.Br utilizou uma metodologia diferente ao do PropBank devido a restrições orçamentárias conforme (47). Os efeitos na qualidade final da anotação devido à essa diferença metodológica, parecem ser até o momento indeterminados. O melhor resultado é a configuração do primeiro experimento.

A Tabela 6.3 mostra a saída do *script* de avaliação *CoNLL Shared Task 2004* para o conjunto de teste. O número de proposições total é de 239 e a percentagem de proposições com os argumentos perfeitamente alocados é de 45,19%. Na coluna à esquerda podemos ver os argumentos, depois temos as seguintes colunas: correto (acertos), excesso (falso positivo), perdas (falso negativo), e as pontuações de precisão e cobertura. O  $F_1$  final foi de 66,23%

Por fim nas Tabelas 6.4 e 6.5 mostram um comparativo entre os sistemas desenvolvidos respectivamente por (20, 41, 22) não apenas quantitativo mas também qualitativo. Na primeira linha temos a pontuação  $F_1$ . A Tabela 6.4 traz os atributos padrão outro, a exceção da posição ID e o *token* FORM dado que ao menos a palavra e a sua posição devem servir de entrada para qualquer sistema de anotação de papéis semânticos. Os demais atributos padrão ouro utilizados para comparação são o LEMMA, MORPH, CTREE (arvore sintagmática) e predicado alvo. Adicionalmente, sistemas que utilizam

	$F_1$		modelo de palavras		LSTM		P.E
	validação	treino	nome	dimensão	camada	tamanho	modelo
1	66,25	88,3	wang2vec	50	bilstm	256x256	crf
2	64,25	90,41	wang2vec	100	bilstm	300x300	crf
3	63,5	88,48	wang2vec	100	bilstm	128x128x128	crf
4	63,43	87,44	wang2vec	50	bilstm	300x300	crf
5	62,77	81,12	wang2vec	50	bilstm	350x350	crf
6	62,59	90,55	wang2vec	100	bilstm	400x400	crf
7	62,58	96,72	word2vec	50	bilstm	350x350	crf
8	62,58	80,62	wang2vec	50	bilstm	100x100x100	crf
9	62,26	79,12	wang2vec	50	bilstm	450x450	crf
10	62,23	75,97	wang2vec	50	bilstm	128x128x128	crf
11	62,01	92,47	wang2vec	50	bilstm	400x400	crf
12	61,93	90,49	wang2vec	50	bilstm	256x256	crf
13	61,9	75,93	wang2vec	100	bilstm	450x450	crf
14	61,61	83,47	wang2vec	100	bilstm	100x100x100	crf
15	61,55	95,63	wang2vec	300	bilstm	350x350	crf
16	61,33	92,04	word2vec	50	bilstm	128x128x128	crf
17	60,6	86,43	wang2vec	50	bilstm	50x50x50	crf
18	60,46	72,15	wang2vec	100	bilstm	350x350	crf
19	60,05	81,7	wang2vec	100	bilstm	64x64x64x64	crf
20	59,45	71,04	wang2vec	50	bilstm	64x64x64x64	crf
21	59,38	93,03	word2vec	50	bilstm	64x64x64x64	crf
22	58,59	72,41	wang2vec	100	bilstm	50x50x50	crf
23	58,3	94,94	wang2vec	300	bilstm	128x128x128	crf
24	57,71	91,01	wang2vec	300	bilstm	64x64x64x64	crf

Tabela 6.2: Resultados etapa 2.

separadores lineares fazem uso de uma série de atributos artificiais baseados nesses atributos padrão outro. Como por exemplo, o caminho do *token* até o predicado ou uma variável indicadora que diz se a sentença está ou não na voz passiva (27). Enquanto atributos manufaturados baseados no *corpus* contém conhecimento de domínio, certos atributos não carregam informação linguística alguma e.g uma variável que diz se o token está em letra maiúscula.

A Tabela 6.5 traz os outros itens tratam-se de sub tarefas intermediárias que compõe alguns dos sistemas. Estas tarefas estão ordenadas de cima para baixo em ordem crescente de complexidade.

Neste trabalho consideramos que o predicado é dado e não precisamos resolver tarefa intermediária alguma, enquanto mantemos um uso de conhecimento específico sobre o idioma baixo.

Número de Proposições: 239  
 Percentagens de proposições perfeitas: 45,19

	corretos	excesso	perdidas	precisão	cobertura	$F_1$
Total	353	173	187	67,11	65,37	66,23
A0	99	16	267	86,09	79,20	82,50
A1	159	53	53	75,00	75,00	75,00
A2	8	13	31	38,10	20,51	26,67
A3	0	0	2	0,00	0,00	0,00
A4	0	2	1	0,00	0,00	0,00
AM-ADV	9	8	11	52,94	45,00	48,65
AM-CAU	1	4	0	20,00	100,00	33,33
AM-DIR	0	0	2	0,00	0,00	0,00
AM-DIS	8	8	3	50,00	72,73	59,26
AM-EXT	0	2	2	0,00	0,00	0,00
AM-LOC	12	14	16	46,15	42,86	44,44
AM-MNR	6	9	13	40,00	31,58	35,29
AM-NEG	16	5	2	76,19	88,89	82,05
AM-PNC	2	4	3	33,33	40,00	36,36
AM-PRD	0	9	7	0,00	0,00	0,00
AM-REC	2	3	3	40,00	40,00	40,00
AM-TMP	31	23	12	57,41	72,09	63,92

Tabela 6.3: Resultados finais.

Atributo   $F_1$	Manchego, 2013(20) 79,20%	Fonseca, 2013(41) 57,25%	Garrido, 2017(22) 58%	Este Trabalho 66,23%
LEMMA	1			
GPOS	1			1
MORPH	1			
CTREE	1	1		
PRED	1			1
MANUFATURADOS	57	1		2

Tabela 6.4: Sistemas por atributos padrão ouro

Atributo   $F_1$	Manchego, 2013(20) 79,20%	Fonseca, 2013(41) 57,25%	Garrido, 2017(22) 58%	Este Trabalho 66,23%
PART-OF-SPEECH	1			1
CHUNKING	1			1
RECONHECIMENTO DE PREDICADO ALVO	1			1
PODA DE SINTÁGMAS	1			
IDENTIFICAÇÃO DE ARGUMENTOS	1	1		1
CLASSIFICAÇÃO DE ARGUMENTOS	1	1		1

Tabela 6.5: Sistemas por sub tarefas

## 7

### Conclusão

O objetivo dessa dissertação é construir um sistema de aprendizado profundo, com baixa dependência de atributos linguísticos e portanto adotamos a arquitetura que se estabeleceu em 2015 como o estado da arte para a tarefa interrompendo um período de 10 anos sem avanços significativos para o idioma do Inglês. Como a arquitetura é menos dependente de conhecimento de domínio, i.e da diferença entre idiomas do Inglês e Português, a expectativa antes do projeto era de que o resultado pudesse superar o estado da arte para o Português. Os resultados experimentais apontam para um comportamento diferente. Entretanto, o sistema foi capaz de superar outras arquiteturas que executam uma *pipeline* de sub tarefas ou que fazem uso de *chunk*.

As contribuições dessa dissertação foram: A adoção de dois modelos de linguagem para o *corpus* em português, *wang2vec* e *word2vec* (56) na tarefa de anotação de papéis semânticos. A proposição de uma arquitetura de aprendizado profundo baseada em redes neurais recorrentes. O teste comparativo entre duas camadas de predição estruturada: CRF e SPN.

O trabalho futuro orientado ao modelo de linguagem é aplicar GloVe (55) *embeddings* e a variante *skipgram* para *wang2vec* e *word2vec*. Uma outra opção seria treinar ELMo *embeddings* (70) para o português dado que o modelo tem mostrado resultados promissores para a resolução da tarefa (35, 39). Um outra possibilidade de melhoria é utilizar o SPN com restrições linguísticas, sem repetição de argumento e.g dois argumentos B-A0. Enquanto o vetor de progresso para a tarefa tem mostrado sistemas com menos atributos sintáticos, o estado da arte (39) re introduz esses atributos. Indicando que o processo de geração automática ainda se beneficia de conhecimento específico – arquiteturas que fazem uso comedido desses atributos têm o potencial de mitigar os efeitos do baixo número de exemplos do PropBank.Br. Uma outra abordagem promissora seria implementar uma rede convolutiva do tipo *gated linear units* (71) que são especializadas em aprender seqüências enquanto são capazes de capturar relacionamentos hierárquicos no texto(34) enquanto sofrem de forma menos severa do problema do gradiente minguante explosivo.

## 8 Referências bibliográficas

- [1] FILLMORE, C. J.. **The case for case**. Holt, Rinehart & Winston, New York, 1968.
- [2] FILLMORE, C. J.. **Frames and the semantics of understanding**. *Quaderni di Semantica*, 6(2):222–254, 1985.
- [3] FILLMORE, C. J.; BAKER, C. F. ; SATO, H.. **Framenet as a ‘net’**. In: *PROCEEDINGS OF LREC*, volumen 4, p. 1091–1094, Lisbon, 2004. ELRA, ELRA.
- [4] R DOWTY, D.. **Thematic proto-roles and argument selection**. *Language*, 67:547–619, 09 1991.
- [5] MÀRQUEZ, L.; CARRERAS, X.; LITKOWSKI, K. C. ; STEVENSON, S.. **Semantic role labeling: An introduction to the special issue**. *Comput. Linguist.*, 34(2):145–159, June 2008.
- [6] LEVIN, B.; SOMERS, H.. **English Verb Classes and Alternations: A Preliminary Investigation**. 1993.
- [7] DANG, H. T.; KIPPER, K.; PALMER, M. ; ROSENZWEIG, J.. **Investigating regular sense extensions based on intersective levin classes**. In: *IN PROCEEDINGS OF COLING-ACL98*, p. 293–299, 1998.
- [8] KOOMEN, P., PUNYAKANOK, V., R. Y.. **Generalized inference with multiple semantic role labeling systems**. In: *CONFERENCE ON COMPUTATIONAL NATURAL LANGUAGE LEARNING (CONLL)*, p. 181–184, 2005.
- [9] ZHOU, JIE, X.. **End-to-end learning of semantic role labeling using recurrent neural networks**. In: *PROC. OF THE ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (ACL)*., 2015.
- [10] CHAMBERS, N.; JURAFSKY, D.. **Template-based information extraction without the templates**. p. 976–986, 01 2011.
- [11] BASTIANELLI, EMANUELE, CASTELLUCCI, GIUSEPPE, C. B.. **Textual inference and meaning representation in human robot interaction**. p. 65–69, 2013.

- [12] SHEN, D.; LAPATA, M.. **Using semantic roles to improve question answering**. In: PROCEEDINGS OF THE 2007 JOINT CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING AND COMPUTATIONAL NATURAL LANGUAGE LEARNING (EMNLP-CONLL), 2007.
- [13] YAN, S.; WAN, X.. **Srrank: Leveraging semantic roles for extractive multi-document summarization**. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22:2048–2058, 12 2014.
- [14] WU, D.; FUNG, P.. **Semantic roles for smt: A hybrid two-pass model**. In: PROCEEDINGS OF HUMAN LANGUAGE TECHNOLOGIES: THE 2009 ANNUAL CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, COMPANION VOLUME: SHORT PAPERS, NAACL-Short '09, p. 13–16, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.
- [15] AZIZ, WILKER, RIOS, MIGUEL, S.. **Shallow semantic trees for smt**. In: PROCEEDINGS OF THE SIXTH WORKSHOP ON STATISTICAL MACHINE TRANSLATION, p. 316–322. Association for Computational Linguistics, 2011.
- [16] LIU, DING, G.. **Semantic role features for machine translation**. 2010.
- [17] HARTMANN, NATHAN SIEGLE, A.. **Anotação automática de papéis semânticos de textos jornalísticos e de opinião sobre árvores sintáticas não revisadas**. Master's thesis, USP, 2017.
- [18] AMANCIO, M. A.; ALUÍZIO, S. M.. **Elaboração textual via definição de entidades mencionadas e de perguntas relacionadas aos verbos em textos simplificados do português**. Master's thesis, USP, 2011.
- [19] HACIOGLU, K.; PRADHAN, S.; WARD, W.; MARTIN, J. H. ; JURAFSKY, D.. **Semantic role labeling by tagging syntactic chunks**. In: IN PROCEEDINGS OF CONLL 2004 SHARED TASK, p. 110–113, 2004.
- [20] MANCHEGO, F. E. A.. **Anotação automática semissupervisionada de papéis semânticos para o português do brasil**. Master's thesis, 2013.
- [21] HE, L.; LEE, K.; LEWIS, M. ; ZETTLEMOYER, L.. **Deep semantic role labeling: What works and what's next**. In: PROCEEDINGS OF

THE ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 2017.

- [22] GARRIDO, LUAN BARBOSA, X.. **Anotação de papés semânticos para o português por *CONDITIONAL RANDOM FIELDS***. Master's thesis, 2017.
- [23] XUE, N.; PALMER, M.. **Calibrating features for semantic role labeling**. In: PROCEEDINGS OF THE 2004 CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING , EMNLP 2004, A MEETING OF SIGDAT, A SPECIAL INTEREST GROUP OF THE ACL, HELD IN CONJUNCTION WITH ACL 2004, 25-26 JULY 2004, BARCELONA, SPAIN, p. 88–94, 2004.
- [24] GILDEA, D.; JURAFSKY, D.. **Automatic labeling of semantic roles**. *Computational Linguistics*, 28:245–288, 2002.
- [25] BAKER, C. F.; FILLMORE, C. J. ; LOWE, J. B.. **The berkeley framenet project**. In: PROCEEDINGS OF THE 36TH ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS AND 17TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL LINGUISTICS - VOLUME 1, ACL '98/COLING '98, p. 86–90, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.
- [26] MARCUS, M. P.; MARCINKIEWICZ, M. A. ; SANTORINI, B.. **Building a large annotated corpus of english: The penn treebank**. *Comput. Linguist.*, 19(2):313–330, June 1993.
- [27] PUNYAKANOK, V, ROTH, D, Y.. **The importance of syntactic parsing and inference in semantic role labeling**. *Computational Linguistics*, 2008.
- [28] COLLOBERT, RONAN, WESTON, JASON, B. K. K. K.. **Natural language processing (almost) from scratch**. 2011.
- [29] SRIVASTAVA, R. K.; GREFF, K. ; SCHMIDHUBER, J.. **Training very deep networks**. CoRR, abs/1507.06228, 2015.
- [30] ZHANG, Y.; CHEN, G.; YU, D.; YAO, K.; KHUDANPUR, S. ; GLASS, J. R.. **Highway long short-term memory rnns for distant speech recognition**. CoRR, abs/1510.08983, 2015.
- [31] SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I. ; SALAKHUTDINOV, R.. **Dropout: A simple way to prevent neural**

- networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014.
- [32] TAN, Z.; WANG, M.; XIE, J.; CHEN, Y. ; SHI, X.. **Deep semantic role labeling with self-attention**. CoRR, abs/1712.01586, 2017.
- [33] VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L. ; POLOSUKHIN, I.. **Attention is all you need**. CoRR, abs/1706.03762, 2017.
- [34] CHENG, J.; DONG, L. ; LAPATA, M.. **Long short-term memory-networks for machine reading**. CoRR, abs/1601.06733, 2016.
- [35] HE, L.; LEE, K.; LEVY, O. ; ZETTLEMOYER, L.. **Jointly predicting predicates and arguments in neural semantic role labeling**. CoRR, abs/1805.04787, 2018.
- [36] LEE, K.; HE, L.; LEWIS, M. ; ZETTLEMOYER, L.. **End-to-end neural coreference resolution**. CoRR, abs/1707.07045, 2017.
- [37] PRADHAN, S.; RAMSHAW, L. A.; MARCUS, M. P.; PALMER, M.; WEISCHDEL, R. M. ; XUE, N.. **Conll-2011 shared task: Modeling unrestricted coreference in ontonotes**. In: PROCEEDINGS OF THE FIFTEENTH CONFERENCE ON COMPUTATIONAL NATURAL LANGUAGE LEARNING: SHARED TASK, CONLL 2011, PORTLAND, OREGON, USA, JUNE 23-24, 2011, p. 1–27, 2011.
- [38] PRADHAN, S.; MOSCHITTI, A.; XUE, N.; URYUPINA, O. ; ZHANG, Y.. **Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes**. In: JOINT CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING AND COMPUTATIONAL NATURAL LANGUAGE LEARNING - PROCEEDINGS OF THE SHARED TASK: MODELING MULTILINGUAL UNRESTRICTED COREFERENCE IN ONTONOTES, EMNLP-CONLL 2012, JULY 13, 2012, JEJU ISLAND, KOREA, p. 1–40, 2012.
- [39] STRUBELL, E.; VERGA, P.; ANDOR, D.; WEISS, D. ; MCCALLUM, A.. **Linguistically-informed self-attention for semantic role labeling**. CoRR, abs/1804.08199, 2018.
- [40] CARUANA, R.. **Multitask learning: A knowledge-based source of inductive bias**. In: PROCEEDINGS OF THE TENTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, p. 41–48. Morgan Kaufmann, 1993.

- [41] FONSECA, ERICK ROCHA. Uma abordagem conexista para anotação de papéis semânticos. Master's thesis, 2013.
- [42] COLLOBERT, R.; WESTON, J.. A unified architecture for natural language processing: Deep neural networks with multitask learning. 2008.
- [43] PALMER, M., GILDEA, D., K.. The proposition bank: An annotated corpus of semantic roles. Computational Linguistics, p. 71–105, March 2005.
- [44] BONIAL, C.; BABKO-MALAYA, O.; CHOI, J.; HWANG, J. ; PALMER, M.. Propbank annotation guidelines. 12 2010.
- [45] CARRERAS, XAVIER, M.. Introduction to the conll-2005 shared task: Semantic role labeling. In: for Computational Linguistics, A., editor, PROCEEDINGS OF THE 9TH CONFERENCE ON COMPUTATIONAL NATURAL LANGUAGE LEARNING (CONLL), p. 152–164, 2005.
- [46] BURCHARDT, A.; ERK, K.; FRANK, A.; KOWALSKI, A. ; PADÓ, S.. Salto—a versatile multi-level annotation tool. 01 2006.
- [47] DURAN, M. S, A.. Propbank-br: a brazilian treebank annotated with semantic role labels. In: EIGHTH INTERNATIONAL CONFERENCE ON LANGUAGE RESOURCES AND EVALUATION, 2012.
- [48] AFONSO, S.; BICK, E.; HABER, R. ; SANTOS, D.. floresta sintá(c)tica”: a treebank for portuguese. In: IN PROCEEDINGS OF THE 3RD INTERN. CONF. ON LANGUAGE RESOURCES AND EVALUATION (LREC, 2002.
- [49] BICK, E.. The Parsing System Palavras Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. Aarhus University Press, 2000.
- [50] DURAN M. S., SEPÚLVEDA-TORRES, L., C. H. A.. Seleção e preparação de sentenças do corpus pln-br para compor o corpus de anotação de papéis semânticos propbank-br.v2., 0124.
- [51] CARRERAS, X.; MÀRQUEZ, L.. Introduction to the conll-2004 shared task: Semantic role labeling. In: PROCEEDINGS OF THE EIGHTH CONFERENCE ON COMPUTATIONAL NATURAL LANGUAGE LEARNING, CONLL 2004, HELD IN COOPERATION WITH HLT-NAACL 2004, BOSTON, MASSACHUSETTS, USA, MAY 6-7, 2004, p. 89–97, 2004.

- [52] BENGIO, Y.; DUCHARME, R.; VINCENT, P. ; JANVIN, C.. **A neural probabilistic language model**. *J. Mach. Learn. Res.*, 3:1137–1155, Mar. 2003.
- [53] RONG, X.. **word2vec parameter learning explained**. *CoRR*, abs/1411.2738, 2014.
- [54] MIKOLOV, T.; CHEN, K.; CORRADO, G. ; DEAN, J.. **Efficient estimation of word representations in vector space**. *CoRR*, abs/1301.3781, 2013.
- [55] PENNINGTON, J.; SOCHER, R. ; MANNING, C. D.. **Glove: Global vectors for word representation**. In: *EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP)*, p. 1532–1543, 2014.
- [56] HARTMANN, N.; FONSECA, E. R.; SHULBY, C.; TREVISO, M. V.; RODRIGUES, J. ; ALUÍSIO, S. M.. **Portuguese word embeddings: Evaluating on word analogies and natural language tasks**. *CoRR*, abs/1708.06025, 2017.
- [57] GOODFELLOW, I.; BENGIO, Y. ; COURVILLE, A.. **Deep Learning**. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [58] HOCHREITER, S.. **Untersuchungen zu dynamischen neuronalen Netzen**. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München, 1991.
- [59] BENGIO, Y.; SIMARD, P. ; FRASCONI, P.. **Learning long-term dependencies with gradient descent is difficult**. *Trans. Neur. Netw.*, 5(2):157–166, Mar. 1994.
- [60] HOCHREITER, S, S.. **Long short term memory**. *Neural Computation*, 9 (8):1735–1780, 1997.
- [61] CRAMMER, K.; SINGER, Y.. **Pranking with ranking**. In: *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 14*, p. 641–647. MIT Press, 2001.
- [62] DOS REIS SILVA, R.. **Extração de citações diretas e indiretas para o português**. Master's thesis, PUC-Rio, 2017.
- [63] MILIDIU, RUY LUIZ, R.. **Structured prediction networks**. 2018.
- [64] SUTTON, C.. **1 an introduction to conditional random fields for relational learning**. 2007.

- [65] ROSENBLATT, F.. **The perceptron: A probabilistic model for information storage and organization in the brain.** *Psychological Review*, p. 65–386, 1958.
- [66] COLLINS, M.. **Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms.** p. 1–8, 2002.
- [67] BELTRÃO, A.. **Anotação de papéis semânticos para o português.** 2016.
- [68] LING, W.; DYER, C.; W BLACK, A. ; TRANCOSO, I.. **Two/too simple adaptations of word2vec for syntax problems.** 05 2015.
- [69] KINGMA, D. P.; BA, J.. **Adam: A method for stochastic optimization.** *CoRR*, abs/1412.6980, 2014.
- [70] PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K. ; ZETTLEMOYER, L.. **Deep contextualized word representations.** *CoRR*, abs/1802.05365, 2018.
- [71] DAUPHIN, Y. N.; FAN, A.; AULI, M. ; GRANGIER, D.. **Language modeling with gated convolutional networks.** *CoRR*, abs/1612.08083, 2016.
- [72] SCHULER, K. K.. **VerbNet: A broad-coverage, comprehensive verb lexicon.** PhD thesis, 2005.
- [73] WEISCHEDEL, R.; HOVY, E.; MARCUS, M.; PALMER, M.; BELVIN, R.; PRADHAN, S.; RAMSHAW, L. ; XUE, N.. **Ontonotes: A large training corpus for enhanced processing.** 01 2011.
- [74] SURDEANU, M.; JOHANSSON, R.; MEYERS, A.; MÀRQUEZ, L. ; NIVRE, J.. **The conll-2008 shared task on joint parsing of syntactic and semantic dependencies.** In: *PROCEEDINGS OF THE TWELFTH CONFERENCE ON COMPUTATIONAL NATURAL LANGUAGE LEARNING, CoNLL '08*, p. 159–177, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.
- [76] HAJIČ, J.; CIARAMITA, M.; JOHANSSON, R.; KAWAHARA, D.; MARTÍ, M. A.; MÀRQUEZ, L.; MEYERS, A.; NIVRE, J.; PADÓ, S.; ŠTĚPÁNEK, J.; STRAÑÁK, P.; SURDEANU, M.; XUE, N. ; ZHANG, Y.. **The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages.** In: *PROCEEDINGS OF THE THIRTEENTH CONFE-*

RENCE ON COMPUTATIONAL NATURAL LANGUAGE LEARNING: SHARED TASK, CoNLL '09, p. 1–18, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

# A

## Apêndice

### A.1

#### O Grafo de Computação

Há várias formas de formalizar um grafo de computação (57). O grafo de computação é um instrumento útil para representar relacionamentos entre variáveis de entrada e saída aplicação de funções. Cada nó no grafo de computação contém uma variável que pode ser um escalar, vetor, matriz ou tensor. Uma operação que é uma aplicação de uma função em uma ou mais variáveis. Sem perda de generalidade essa função, retorna apenas uma variável. Se a variável  $y$  é computada por uma aplicação de  $x$  então uma aresta é desenhada entre  $x$  e  $y$ . Algumas variáveis podem ser rotuladas com nomes de funções, como no caso de operações com mais de uma variável de entrada. Variáveis podem ser nomeadas ou não. Arestas podem ser rotuladas ou não. Na Figura A.1 são dois grafos distintos que representam as operações (a)  $z = xy$  (b)  $y = \sigma(\mathbf{w}^T \mathbf{x} + \mathbf{b})$

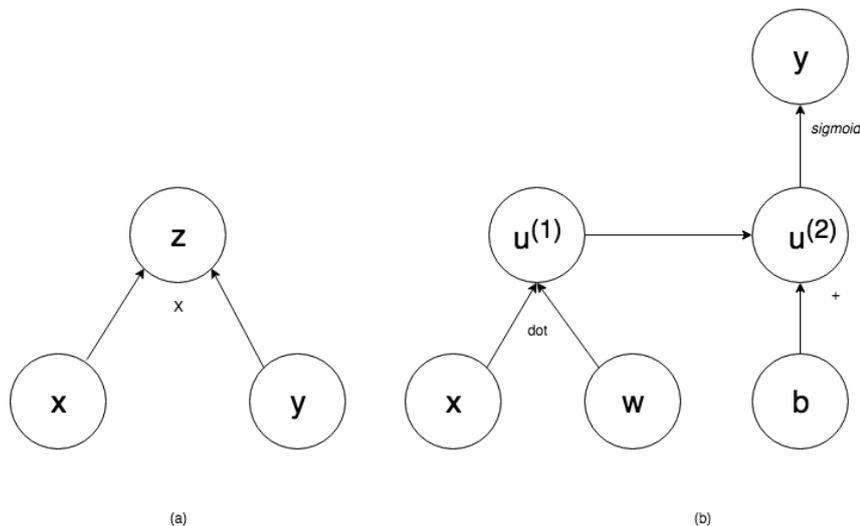


Figura A.1: O gráfico de computação.

No caso das RNRs é útil definir **desdobramento** (*unfolding*) de uma computação recursiva, que tem uma estrutura repetitiva correspondendo à uma cadeia de eventos, e que divide os seus parâmetros  $\theta$  através do tempo. A equação A-1 define um sistema dinâmico clássico, ela é recursiva porque  $\mathbf{s}$

no tempo  $t$  se refere a ela própria no tempo  $t - 1$ . Para um número finito  $\tau$ , o grafo pode ser desdobrado  $\tau - 1$  vezes, A.1. À esquerda da Figura A.2 um quadrado representado um passo no tempo, à direita está o seu equivalente desdobrado, ambas representam a Equação A-1 Os parâmetros  $\theta$  são divididos no tempo.

$$\mathbf{s}^{(t)} = f(\mathbf{s}^{(t-1)}; \theta) \tag{A-1}$$

$$\begin{aligned} \mathbf{s}^{(4)} &= f(\mathbf{s}^{(3)}; \theta) \\ &= f(f(\mathbf{s}^{(2)}; \theta); \theta) \\ &= f(f(f(\mathbf{s}^{(1)}; \theta); \theta); \theta) \end{aligned}$$

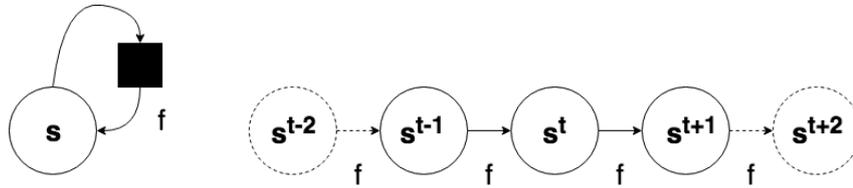


Figura A.2: Desdobramento do grafo de computação.