Anotação automática semissupervisionada de papéis semânticos para o português do Brasil

Fernando Emilio Alva Manchego

Anotação automática semissupervisionada de papéis semânticos para o português do Brasil

Fernando Emilio Alva Manchego

Orientador: **Prof. Dr. João Luís Garcia Rosa**

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

USP – São Carlos
Janeiro de 2013

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

A472a

Alva-Manchego, Fernando Emilio
Anotação automática semissupervisionada de papéis semânticos para o português do Brasil / Fernando Emilio Alva-Manchego; orientador João Luís Garcia Rosa. -- São Carlos, 2013.
137 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2013.

1. Anotação de Papéis Semânticos. 2. Aprendizado Semissupervisionado. 3. Self-training. 4. Processamento de Língua Natural. 5. Linguística Computacional. I. Rosa, João Luís Garcia, orient. II. Título.

Agradecimentos

À minha família, pelo seu amor, sua confiança e seu apoio em todos os caminhos que me proponho seguir. Sem eles não estaria aqui, confiante de que posso enfrentar qualquer desafio que se apresente. Sei que sempre posso contar com vocês, sem importar a distância que nos separe.

Ao meu orientador, o Prof. João Rosa, pela oportunidade de pesquisar em uma área muito interessante do processamento de língua natural, e a confiança depositada no meu trabalho durante o mestrado.

À Profa. Mirella Lapata, minha supervisora durante o estágio na Universidade de Edimburgo, pela orientação e conselhos sobre como realizar boa pesquisa na área, e as palavras motivadoras quando parecia que o estágio não cumpriria com os objetivos propostos.

Aos membros do NILC, professores e alunos, pelas conversas, cafezinhos, festas, *happy hours* e, em geral, todos os momentos de convivência. Obrigado por terem compartilhado comigo a sua motivação e determinação por realizar pesquisa em uma área tão desafiadora como é PLN.

Aos "nilcenses" e os meus amigos do ICMC, aqueles que conheci durante as aulas ou nas horas de lazer. Muito obrigado por terem sido minha família no Brasil, e terem me ajudado a emadurecer e me tornar uma melhor pessoa.

À FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo) pelo apoio financeiro durante o mestrado e o estágio no exterior.

Resumo

A anotação de papéis semânticos (APS) é uma tarefa do processamento de língua natural (PLN) que permite analisar parte do significado das sentenças através da detecção dos participantes dos eventos (e dos eventos em si) que estão sendo descritos nelas, o que é essencial para que os computadores possam usar efetivamente a informação codificada no texto. A maior parte das pesquisas desenvolvidas em APS tem sido feita para textos em inglês, considerando as particularidades gramaticais e semânticas dessa língua, o que impede que essas ferramentas e resultados sejam diretamente transportáveis para outras línguas como o português. A maioria dos sistemas de APS atuais emprega métodos de aprendizado de máquina supervisionado e, portanto, precisa de um *corpus* grande de sentenças anotadas com papéis semânticos para aprender corretamente a tarefa. No caso do português do Brasil, um recurso lexical que provê este tipo de informação foi recentemente disponibilizado: o PropBank.Br. Contudo, em comparação com os *corpora* para outras línguas como o inglês, o corpus fornecido por este projeto é pequeno e, portanto, não permitiria que um classificador treinado supervisionadamente realizasse a tarefa de anotação com alto desempenho. Para tratar esta dificuldade, neste trabalho emprega-se uma abordagem semissupervisionada capaz de extrair informação relevante tanto dos dados anotados disponíveis como de dados não anotados, tornando-a menos dependente do *corpus* de treinamento. Implementa-se o algoritmo *self-training* com modelos de regressão logística (ou máxima entropia) como classificador base, para anotar o *corpus* Bosque (a seção correspondente ao CETENFolha) da Floresta Sintá(c)tica com as etiquetas do PropBank.Br. Ao algoritmo original se incorpora balanceamento e medidas de similaridade entre os argumentos de um verbo específico para melhorar o desempenho na tarefa de classificação de argumentos. Usando um *benchmark* de avaliação implementado neste trabalho, a abordagem semissupervisonada proposta obteve um desempenho estatisticamente comparável ao de um classificador treinado supervisionadamente com uma maior quantidade de dados anotados (80,5 vs. 82,3 de F_1, $p > 0,01$).

Palavras-chave: Anotação de papéis semânticos. Aprendizado semissupervisionado. Processamento de língua natural.

Abstract

Semantic role labeling (SRL) is a natural language processing (NLP) task able to analyze part of the meaning of sentences through the detection of the events they describe and the participants involved, which is essential for computers to effectively understand the information coded in text. Most of the research carried out in SRL has been done for texts in English, considering the grammatical and semantic particularities of that language, which prevents those tools and results to be directly transported to other languages such as Portuguese. Most current SRL systems use supervised machine learning methods and require a big *corpus* of sentences annotated with semantic roles in order to learn how to perform the task properly. For Brazilian Portuguese, a lexical resource that provides this type of information has recently become available: PropBank.Br. However, in comparison with *corpora* for other languages such as English, the *corpus* provided by that project is small and it wouldn't allow a supervised classifier to perform the labeling task with good performance. To deal with this problem, in this dissertation we use a semi-supervised approach capable of extracting relevant information both from annotated and non-annotated data available, making it less dependent on the training *corpus*. We implemented the self-training algorithm with logistic regression (or maximum entropy) models as base classifier to label the *corpus* Bosque (section CETENFolha) from the Floresta Sintá(c)tica with the PropBank.Br semantic role tags. To the original algorithm, we incorporated balancing and similarity measures between verb-specific arguments so as to improve the performance of the system in the argument classification task. Using an evaluation benchmark implemented in this research project, the proposed semi-supervised approach has a statistical comparable performance as the one of a supervised classifier trained with more annotated data (80,5 vs. 82,3 de F_1, $p > 0,01$).

Keywords: Semantic role labeling. Semi-supervised learning. Natural language processing.

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Sumário

Capítulo

1

Introdução

1.1 Contextualização e Motivação

Em toda a variedade de sistemas de Processamento de Língua Natural (PLN) que existem, os encarregados de busca e recuperação de documentos ou informações a partir de padrões textuais são populares atualmente, dada a imensa quantidade de informação veiculada na web (Strube de Lima et al., 2007). Esses sistemas enfrentam um problema importante na hora de processar a entrada do usuário: a ambiguidade do significado do texto de entrada (Rosa, 2008). O desafio consiste em realmente entender a mensagem do texto, diferenciando-a de qualquer outra interpretação possível; isto é, compreender seu significado. A subárea do PLN encarregada dessa tarefa é a Análise Semântica.

Existem várias pesquisas em análise semântica realizadas por diferentes grupos de pesquisa em universidades reconhecidas internacionalmente (CMU[1], Stanford[2], Cambridge[3], Edinburgh[4], etc.), assim como em institutos de pesquisa de grandes empresas (Google[5], Microsoft[6], etc.). Porém, a maior densidade de produtos e resultados se concentra na língua inglesa, e estes resultados não são diretamente transportáveis para outras línguas como o português (Strube de Lima et al., 2007). Portanto, é evidente que a comunidade de PLN que trabalha com o português precisa desenvolver mais pesquisas em análise semântica de textos nesta língua.

[1] http://www.lti.cs.cmu.edu/research/projects.shtml
[2] http://nlp.stanford.edu/research.shtml
[3] http://www.cl.cam.ac.uk/research/nl/projects/
[4] http://www.ilcc.inf.ed.ac.uk/research/research-in-ilcc
[5] http://research.google.com/pubs/NaturalLanguageProcessing.html
[6] http://research.microsoft.com/en-us/groups/nlp/

Uma forma de entender o significado (semântica) de uma sentença é analisando como se relacionam os constituintes da mesma; em particular, como é que o verbo determina o comportamento dos demais constituintes. Ao se perguntar ao verbo `quem?`, `o quê?`, `para quem?`, `quando?` e `onde?`, podem-se obter as respostas a estas questões dos outros constituintes da sentença (sujeito, objeto direto, objeto indireto e modificadores), o que é possível porque existem relações conceituais entre estes constituintes e o verbo. A sentença, na sua estrutura básica, consiste de um verbo e de um ou mais sintagmas nominais, cada um associado com o verbo em uma relação particular (Fillmore et al., 1968). As relações semânticas entre o verbo e os seus argumentos (os outros constituintes da sentença) recebem o nome de **papéis semânticos**. A tarefa de identificar quais grupos de palavras (ou sintagmas) atuam como os argumentos de um determinado verbo é chamada de **anotação de papéis semânticos** (APS) (Shamsfard e Mousavi, 2008).

A APS permite detectar aspectos dos eventos que estão sendo descritos na sentença, assim como os participantes dos mesmos, o que é essencial para que os computadores possam usar efetivamente a informação codificada em texto (Palmer et al., 2010). Devido ao nível de análise de textos que a APS fornece, esta tem aplicações em muitas áreas de PLN como extração de informação (Surdeanu et al., 2003; Moreda et al., 2007), sistemas de perguntas e respostas (Stenchikova et al., 2006; Shen e Lapata, 2007; Frank et al., 2007; Stoyanchev et al., 2008), sumarização automática (Melli et al., 2005; Suanmali et al., 2010) e tradução automática (Giménez e Màrquez, 2007, 2008; Wu e Fung, 2009a,b).

Para o inglês, existem principalmente três recursos lexicais que fornecem informação sobre papéis semânticos: **FrameNet** (Baker et al., 1998), baseada na Semântica de *Frames* de Fillmore (Fillmore, 1985) e que utiliza etiquetas de papéis semânticos mais refinadas chamadas de *frame elements*; **VerbNet** (Kipper-Schuler, 2005), um léxico computacional de verbos construído com uma abordagem baseada nas classes de Levin (Levin, 1993) e que permite explicitar uma relação entre sintaxe e semântica; e **PropBank** (Palmer et al., 2005), um *corpus* anotado com papéis semânticos específicos para cada verbo, criado visando o treinamento de sistemas baseados em aprendizado de máquina (AM).

Para anotar automaticamente, a maioria dos sistemas de APS atuais emprega técnicas de AM para realizar a tarefa, uma vez que esta pode ser considerada como um problema de classificação: considerando um verbo e cada constituinte de uma árvore sintática, seleciona-se, de um conjunto pré-definido, as etiquetas semânticas para cada constituinte em relação ao verbo (Palmer et al., 2010). Para treinar o classificador encarregado da anotação, extraem-se atributos dos constituintes das sentenças que capturam aspectos sintáticos e léxico-semânticos relevantes para, entre outras coisas, detectar o fenômeno de alternância sintática[7] e atribuir a etiqueta de papel semântico mais apropriada.

[7]Ver uma explicação mais detalhada na Seção 2.2

Para treinar o sistema de AM que predirá as etiquetas, Palmer et al. (2010) indicam que a experiência em APS confirma que abordagens discriminativas, como *Support Vector Machines* (Johansson e Nugues, 2006; Pradhan et al., 2005, 2008) e Máxima Entropia (Fleischman et al., 2003; He e Gildea, 2007; Zadeh Kaljahi, 2010) são mais adequadas para explorar um grande número de atributos do que modelos baseados em frequência como árvores de decisão (Surdeanu et al., 2003), que rapidamente sofrem pelo espalhamento dos dados devido ao particionamento dos mesmos na combinação de atributos.

A comunidade de PLN em português mostra um crescente interesse em desenvolver pesquisa sobre análise semântica de textos nesta língua. Ênfase está sendo dada na criação de recursos lexicais que possam fornecer os dados anotados necessários para a implementação de sistemas baseados em AM. Por exemplo, o projeto de Scarton e Aluísio (2012) propõe um método semiautomático para a criação da VerbNet.Br, baseado nos mapeamentos existentes entre a VerbNet e a WordNet.Pr[8], e os alinhamentos entre a WordNet.Pr e a WordNet.Br (Dias-da-Silva, 2004; Dias-da-Silva et al., 2006). Por outro lado, o PropBank.Br (Duran e Aluísio, 2012), construído usando a metodologia de PropBank do inglês, pode fornecer os dados de treinamento necessários para qualquer sistema de APS automática, empregando AM, que considere o uso do conjunto de etiquetas semânticas empregadas nesse projeto. Foram disponibilizados dados anotados deste *corpus*, e pesquisas em APS usando este recurso têm sido propostas (Alva-Manchego e Rosa, 2012b; Fonseca e Rosa, 2012) embora ainda sem publicar algum resultado obtido.

A maioria dos sistemas de APS atuais corresponde a pesquisas realizadas para o inglês, inviabilizando seu uso direto para outras línguas como o português. Este fato, ao invés de representar uma desvantagem, serve como motivação para promover a pesquisa nesta tarefa que demonstra ser útil em uma grande variedade de aplicações de PLN. Os trabalhos realizados para o inglês servem como base para identificar quais caminhos percorrer na pesquisa de APS para o português, mas não a limitam, uma vez que ainda existem várias abordagens não exploradas. Em particular, a grande maioria de sistemas para APS em inglês foi desenvolvida empregando aprendizado supervisionado porque têm disponíveis recursos lexicais com essa informação que podem ser usados como dados de treinamento e teste. Embora este método permita obter bons resultados, sofre de problemas de (in)dependência de domínio e escala.

No caso do português do Brasil, os projetos para criar os recursos lexicais que disponibilizam *corpus* anotados úteis para sistemas estatísticos não possuem uma extensão tão significativa para um apropriado aprendizado supervisionado. O *corpus* PropBank.Br foi desenvolvido com base em um *corpus* pequeno (aprox. 180 mil palavras), em com-

[8]WordNet de Princeton (WN.Pr), desenvolvida para o inglês norte-americano, e com base estudos aplicados por pesquisadores do Laboratório de Ciências Cognitivas da Universidade de Princeton, Estados Unidos (Fellbaum, 1998)

paração com o *corpus* base do PropBank original (aprox. 1 milhão de palavras). Além disso, a distribuição das sentenças anotadas por verbo no *corpus* é muito desbalanceada. Como pode-se ver na Fig. 1.1, quase 70% de verbos no *corpus* possuem, no máximo, quatro instâncias anotadas. Um sistema baseado em métodos supervisionados de AM teria dificuldades para generalizar apropriadamente e atingir um aprendizado bem sucedido.

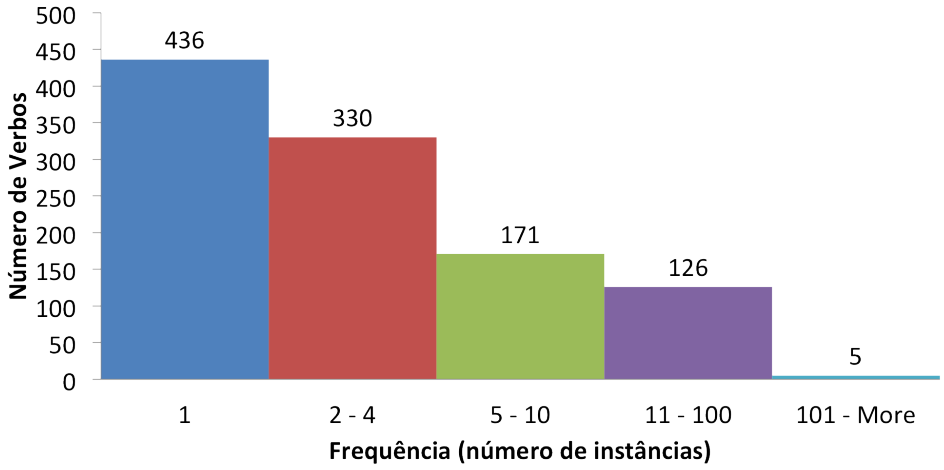


Figura 1.1: Distribuição do número de instâncias anotadas por verbo alvo no *corpus* PropBank.Br.

Para resolver o problema de escassez de dados anotados, têm sido pesquisadas abordagens não supervisionadas (Lang e Lapata, 2010, 2011a,b; Abend et al., 2009; Abend e Rappoport, 2010) e semissupervisionadas (Fürstenau e Lapata, 2009a,b, 2012; Zadeh Kaljahi, 2010), que visam aproveitar atributos dos dados não anotados no aprendizado do sistema de anotação. Esta última abordagem é utilizada na implementação do anotador para o português do Brasil, porque permite aproveitar os dados do *corpus* PropBank.Br, e analisar como aproveitar a informação fornecida por dados não anotados para compensar o desbalanceamento do *corpus*. Dessa maneira, o sistema é capaz de anotar sentenças com características que não foram necessariamente encontradas nos dados de treinamento.

1.2 Hipótese e Objetivos

Considerando a motivação apresentada na seção anterior, a hipótese deste trabalho é que **é possível empregar técnicas de aprendizado de máquina semissupervisionado para anotar automaticamente com papéis semânticos sentenças escritas em português do Brasil com um desempenho comparável ao de um anotador supervisionado para a mesma língua**.

O objetivo principal deste trabalho, portanto, consistiu em usar atributos lexicais, sintáticos e/ou semânticos de sentenças em português do Brasil para treinar um classificador de forma semissupervisionada que fosse capaz de anotar automaticamente estas sentenças com papéis semânticos. Os objetivos específicos perseguidos no desenvolvimento deste sistema são:

- Identificar quais atributos lexicais e sintáticos das sentenças em português beneficiam a anotação dos seus constituintes com papéis semânticos.
- Criar um *baseline* que permita avaliar e/ou comparar o desempenho de sistemas de anotação de papéis semânticos para o português.
- Explorar técnicas não supervisionadas que indiquem como melhor aproveitar a informação fornecida pelos dados não anotados.
- Treinar um classificador semissupervisionadamente a partir das sentenças do *corpus* PropBank.Br, que use os atributos identificados previamente e aproveite a informação fornecida pelos dados não anotados.
- Disponibilizar um anotadores automáticos de papéis semânticos que sirvam como sistemas de referência para a pesquisa na área na língua portuguesa.

1.3 Organização da Monografia

No Capítulo 2, apresenta-se a noção de papel semântico dada por Charles Fillmore, e como ela evoluiu desde a Gramática de Casos até a Semântica de *Frames*. Também descreve-se o fenômeno de alternância sintática (de diátese) e como ele faz da anotação de papéis semânticos uma tarefa desafiadora e útil. Além disso, são descritos os principais recursos lexicais disponíveis para a língua inglesa e seus equivalentes para o português.

No Capítulo 3, analisam-se as principais abordagens computacionais existentes para anotação automática de papéis semânticos e descrevem-se sistemas representativos dessas abordagens. Especificam-se os atributos que são extraídos dos constituintes das sentenças, os métodos mais empregados e como é feita a avaliação dos resultados obtidos por esses sistemas. Também, apresentam-se algumas pesquisas na área realizadas para o português.

No Capítulo 4, apresenta-se um *benchmark* implementado para avaliar e comparar sistemas de anotação de papéis semânticos para o português do Brasil. Usando os recursos fornecidos pelo *benchmark*, implementou-se um sistema de anotação supervisionado, que permite avaliar um conjunto de atributos da literatura e sua aplicação para o português, assim com servir de referência para comparação com o sistema semissupervisionado desenvolvido neste trabalho.

No Capítulo 5, detalha-se a implementação e adaptação para o português de três métodos não supervisionados para indução de papéis semânticos. O estudo destes métodos baseados em particionamento de grafos de similaridade permite entender como melhor aproveitar a informação fornecida por dados não anotados.

No Capítulo 6, descreve-se a implementação de um sistema semissupervisionado de anotação de papéis semânticos baseado no algoritmo *self-training* e medidas de similaridade sintático-lexicais entre os dados anotados e os não anotados. Este sistema, aproveitando a informação fornecida pelos dados não anotados, atinge um melhor desempenho que um sistema supervisionado treinado no (pequeno) conjunto de dados anotados.

Finalmente, no Capítulo 7 são apresentadas as conclusões obtidas deste trabalho, detalhando as contribuições realizadas na área de pesquisa e possíveis trabalhos futuros.

Capítulo

2

Papéis Semânticos: Teorias Linguísticas e Recursos Lexicais

O desenvolvimento de aplicações na área de Processamento de Língua Natural (PLN), devido ao seu caráter multidisciplinar (linguística e computação), deve-se iniciar com um estudo dos formalismos linguísticos básicos correspondentes à tarefa que se deseja realizar (Dias-da-Silva, 1996). Assim, inicia-se esta monografia com uma revisão dos conceitos linguísticos envolvidos na tarefa de anotação de papéis semânticos (APS). Além disso, apresenta-se uma descrição dos principais recursos lexicais disponíveis com informação sobre papéis semânticos (alguns deles motivados por alguma teoria linguística particular).

Neste capítulo apresenta-se a noção de papel semântico de Fillmore e como esta foi evoluindo desde a Gramática de Casos até a Semântica de *Frames* (Seção 2.1). Também, apresenta-se o estudo de Levin sobre classes verbais e as suas alternâncias sintáticas e como este fenômeno linguístico torna a APS uma tarefa desafiadora (Seção 2.2). Além disso, são descritos alguns recursos lexicais que fornecem conjuntos de etiquetas semânticas para serem usados na anotação de dados (Seção 2.3) e, por último, apresentam-se algumas considerações finais (Seção 2.4).

2.1 Noção de Papel Semântico

Uma parte do significado de uma sentença em língua natural como:

João quebrou a janela com a pedra. (2.1)

pode ser analisado identificando o evento descrito na sentença indicado pelo verbo quebrar e as entidades referidas no evento por João, a janela e a pedra. Além disso, cada entidade pode ser representada linguísticamente em termos de um **papel semântico**, que descreve a forma em que esta entidade está envolvida no evento. Por exemplo, João pode ser caracterizado como a entidade que realiza a ação, i.e., o `AGENTE`, enquanto a janela seria a entidade afetada pela ação, i.e., o `PACIENTE`. Esta noção de papel semântico foi dada nas teorias de Charles Fillmore que são apresentadas a seguir.

2.1.1 Gramática de Casos

A Gramática de Casos (Fillmore et al., 1968) é uma teoria para representação semântica baseada nas relações existentes entre a ação (ou estado) denotada pelo verbo e seus argumentos. Essas relações são chamadas de **relações de casos**, ou, simplesmente, **casos conceituais**.

A afirmação principal da teoria de Fillmore é que a sentença, na sua estrutura básica, consiste de um verbo e de um ou mais sintagmas nominais, cada um associado com o verbo em uma relação de caso particular, e que cada relação de caso acontece só uma vez em uma sentença simples. Então, estabelecem-se tipos de sentenças de acordo com as várias combinações possíveis de casos e pode-se classificar os verbos de acordo com o *case frame* em que possam ser inseridos (Lima, 1982).

Fillmore parte da hipótese de que as línguas humanas são restritas, de modo que as relações entre os constituintes de uma sentença se enquadram em um pequeno número de tipos, os quais caracterizam, então, os chamados casos conceituais. Estes podem ser identificados por julgamentos que os seres humanos fazem acerca dos acontecimentos que ocorrem ao seu redor. Em Fillmore et al. (1968) propõem-se, para o contexto de mundo considerado, seis casos conceituais:

1. `AGENTIVO (A)`: o caso do instigador animado perceptivo da ação identificada pelo verbo.

2. `INSTRUMENTAL (I)`: o caso da força ou objeto inanimado causalmente envolvido na ação ou estado identificado pelo verbo.

3. `DATIVO (D)`: o caso do ser animado afetado pelo estado ou ação identificado pelo verbo.

4. `FACTITIVO (F)`: o caso do objeto ou ser resultante da ação ou estado identificado pelo verbo ou compreendido como parte do significado do verbo.

5. `LOCATIVO (L)`: o caso que identifica a localização ou orientação espacial do estado ou ação identificado pelo verbo.

6. `OBJETIVO (O)`: o caso mais neutro semanticamente, o caso de qualquer coisa representada por um substantivo cujo papel na ação ou estado identificados pelo verbo é determinado pela interpretação semântica do próprio verbo.

O desenvolvimento e sucessivas modificações ao modelo de Fillmore têm modificado a lista de casos original de 1968. A seguir são apresentados os casos que correspondem à versão de 1971 (Cook, 1989).

1. `AGENTE (A)`: instigador da ação, a principal causa do evento.
2. `EXPERIENCIADOR (E)`: inclui a maioria das funções do `DATIVO`, mas exclui os verbos não psicológicos de mudança de estado como morrer e crescer.
3. `INSTRUMENTO (I)`: a causa imediata de um evento. Se `AGENTE` e `INSTRUMENTO` co-ocorrem, o `AGENTE` é o instigador do evento e o `INSTRUMENTO` é a causa mais imediatamente em contato com o evento.
4. `OBJETO (O)`: o caso mais neutro, a entidade que se move ou sofre mudança.
5. `FONTE (S)`: é a origem ou ponto de partida do movimento; refere-se principalmente ao lugar desde o qual o movimento começa.
6. `META (G)`: é o ponto final do movimento; refere-se ao lugar para o qual o movimento tende.
7. `LOCAL (L)`: o lugar onde o objeto ou evento está localizado.
8. `TEMPO (T)`: momento em que um objeto ou evento está localizado.
9. `BENEFACTIVO (B)`: aquele beneficiado pelo evento ou atividade.

Em suma, a Gramática de Casos é uma teoria que trata sobre a semântica das sentenças; não se preocupa com a semântica do discurso ou das palavras. Lida só com a estrutura interna das orações e, até mesmo dentro de uma oração, não lida com todos os elementos de significado; só com a estrutura essencial do predicado (o verbo). Esta teoria tenta analisar o significado de uma oração em termos de um predicado central e os argumentos requeridos por ele, assim como identificar os papéis semânticos destes argumentos.

2.1.2 Semântica de Frames

Um dos questionamentos à Gramática de Casos é sobre o conjunto de etiquetas de papéis semânticos ou se, de fato, é possível caracterizar os predicados das línguas naturais

usando um conjunto pequeno de tais etiquetas. Considerando estas limitações, e com base na noção de *frames* da área de Representação do Conhecimento na Inteligência Artificial (Minsky, 1975)[1], os próximos trabalhos de Fillmore e colaboradores (Fillmore, 1976, 1982, 1985; Fillmore et al., 2003) levaram à convicção de que um pequeno conjunto fixo de papéis de caso não era suficiente para caracterizar as propriedades de complementação dos itens lexicais.

Assim, foi formulada a **Semântica de *Frames*** como uma abordagem para o estudo do significado lexical. A ideia central desta teoria é que os significados das palavras são melhor compreendidos com referência às estruturas conceituais que as suportam e motivam, chamadas de ***frames* semânticos**.

O termo *frame* faz referência a qualquer sistema de conceitos relacionados de tal forma que, para entender um deles, é necessário entender toda a estrutura correspondente. Quando algum destes elementos da estrutura é introduzido em um texto, todos os demais estão disponíveis automaticamente. Por exemplo, considere-se o *frame* `TRANSAÇÃO COMERCIAL`: os membros desse *frame* são os indivíduos e acessórios que participam nessas transações (chamados de ***frame elements***). Nesse caso, os indivíduos são os protagonistas da transação, e os acessórios são os objetos que sofrem alteração de propriedade, um deles sendo o dinheiro.

Palavras ou frases **evocam** *frames* particulares ou **instanciam** elementos particulares desses *frames*. Por exemplo, se é examinado o *frame* `TRANSAÇÃO COMERCIAL`, será necessário identificar *frame elements* como `COMPRADOR`, `VENDEDOR`, `PAGAMENTO`, `BENS`, etc., e pode-se dizer que palavras como comprar, vender, pagar, cliente, etc., são capazes de evocar este *frame*. Igualmente, em algumas sentenças será possível encontrar sintagmas como João, o cliente instanciando `COMPRADOR`, ou um carro novo instanciando `BENS`.

A Semântica de *Frames* encontra-se incorporada na FrameNet (Baker et al., 1998), que é um recurso lexical com descrições de *frames* semânticos de vários milhares de itens lexicais do inglês. Estas descrições estão baseadas nas anotações semânticas manuais (feitas por linguistas e lexicógrafos) de sentenças extraídas de *corpora* de textos e na análise sistemática dos padrões semânticos que elas exemplificam. A FrameNet é descrita com mais detalhes na Seção 2.3.1.

2.2 Classes de Verbos de Levin

Levin (1993) apresenta um estudo de cerca de 3.000 verbos do inglês e as suas alternâncias sintáticas, para agrupá-los em classes dentro das quais os verbos possuem

[1]Para Minsky, os *frames* representam situações arbitrárias (p.e., comer em uma mesa, um processo no tribunal, uma campanha eleitoral) e são porções de conhecimento que ajudam a entender instâncias específicas das situações que descrevem.

comportamento e significado compartilhados. O trabalho de Levin supõe que o comportamento de um verbo, particularmente com respeito à expressão e interpretação dos seus argumentos, está determinado pelo seu significado. Assim, o comportamento dos verbos pode ser usado para investigar aspectos linguisticamente relevantes ao seu significado.

Os verbos, como elementos que possuem argumentos, apresentam conjuntos de propriedades especialmente complexos. Os falantes nativos de uma língua podem realizar julgamentos extremamente sutis com respeito à ocorrência de verbos com uma gama de possíveis combinações de argumentos e adjuntos em várias expressões sintáticas. Por exemplo, os falantes sabem em quais **alternâncias de diátese** – alternâncias nas expressões de argumentos, algumas vezes acompanhadas de mudança de significado – os verbos podem participar.

Por exemplo, um falante da língua sabe se um verbo pode participar em uma ou várias alternâncias de transitividade – alternâncias de diátese que envolvem uma mudança na transitividade do verbo. Assim, por exemplo, embora o verbo quebrar apresente usos transitivos e intransitivos, esta possibilidade não está disponível para o verbo cortar.

a. João quebrou a janela com a pedra.
b. A janela quebrou. (2.2)

a. Maria cortou o tecido com uma tesoura.
b. *O tecido cortou. (2.3)

Segundo Levin, o que permite que um falante de uma língua determine o comportamento de um verbo é o seu significado. Provavelmente, previsões sobre o comportamento de um verbo são possíveis porque certas propriedades sintáticas estão associadas com verbos de um determinado tipo semântico.

Através do estudo das alternâncias de diátese dos verbos *break* (quebrar), *cut* (cortar), *hit* (bater) e *touch* (tocar), Levin mostra que os verbos em inglês (e em outras línguas) se agrupam em **classes** que compartilham componentes de significado. Os membros de uma classe têm em comum uma gama de propriedades, incluindo as possíveis expressões e interpretações dos seus argumentos.

As classes de verbos são definidas baseadas na habilidade de cada verbo de ocorrer ou não ocorrer em pares de *frames* sintáticos que preservam o significado (alternâncias de diátese). De acordo com esta teoria, os membros de uma classe devem compartilhar um ou mais componentes semânticos que são preservados da mesma forma.

Levin organiza aproximadamente 3.100 verbos do inglês em 48 classes principais, as quais são logo subdivididas em classes menores e mais específicas, totalizando 192. Verbos com mais de um sentido (aproximadamente 784) aparecem em mais de uma classe. Na Tabela 2.1 apresentam-se alguns exemplos das classes de Levin, com seus membros,

algumas alternâncias características e os componentes semânticos subjacentes sugeridos.

Tabela 2.1: Exemplos de classes de Levin. Fonte: Palmer et al. (2010)

Classe **break 45.1**	
Frames Sintáticos	*John broke the jar.* (João quebrou o vaso.) *The jar broke.* (O vaso quebrou) *Jars break easily.* (Vasos quebram facilmente.)
Membros	*break, chip, crack, crash, crush, fracture, rip, shatter, smash, snap, splinter, snip, tear*
Componentes Semânticos	mudança de estado
Classe **cut 21.1**	
Frames Sintáticos	*John cut the bread.* (João cortou o pão.) **The bread cut.* (*O pão cortou.) *Bread cuts easily.* (O pão corta fácil.)
Membros	*chip, chop, clip, cut, hack, hey, rip, saw, scrape, scratch, slah, slice, snip*
Componentes Semânticos	mudança de estado, ação reconhecível, instrumento afiado
Classe **hit 18.1**	
Frames Sintáticos	*John hit the wall.* (João bateu na parede.) **The wall hit.* (*A parede bateu.) **Walls hit easily.* (*Paredes batem facilmente.)
Membros	*bang, bash, click, dash, squash, tamp, thump, thwack, whack, batter, beat, bump, butt, drum, hammer, hit, jab, kick, knock, lash, pound, rap, slap, anack, smash, strike, tap*
Componentes Semânticos	contato, exercício de força

Esta classificação de verbos pode parecer não guardar relação com a análise da noção de papéis semânticos, porque não fala sobre eles explicitamente. Contudo, os papéis semânticos são referidos implicitamente pela natureza das alternâncias de diátese, que são definidas como preservadoras de significado. Este tipo de alternância refere-se a mudança de função sintática dos argumentos do verbo, mas carregando consigo os seus

papéis semânticos. Por exemplo, para o predicado quebrar:

a. [João AGENTE] **quebrou** [a janela PACIENTE] com [a pedra INSTRUMENTO].

b. [A pedra INSTRUMENTO] **quebrou** [a janela PACIENTE]. (2.4)

c. [A janela PACIENTE] **quebrou**.

em 2.4a, a pedra é o objeto indireto, enquanto em 2.4b é o sujeito; contudo, em ambas sentenças, a pedra ainda possui o papel de `INSTRUMENTO`. O mesmo acontece com a janela em 2.4a e 2.4c: em ambas possui o papel de `PACIENTE`, embora seja o objeto direto e o sujeito, respectivamente.

2.3 Recursos Lexicais Disponíveis

Existem recursos linguísticos que disponibilizam dados anotados seguindo as ideias propostas por algumas das teorias apresentadas na seção anterior. Esse é o caso da **FrameNet** (Baker et al., 1998), que segue a Semântica de *Frames* de Fillmore, e da **VerbNet** (Kipper-Schuler, 2005) para o caso das Classes Verbais de Levin. Por sua vez, o **PropBank** (Palmer et al., 2005) considera-se teoricamente neutro e está mais focado em fornecer dados para o treinamento de sistemas baseados em aprendizado de máquina. Nesta seção, estes três recursos são descritos, assim como os projetos que visam criar recursos lexicais equivalentes para o português do Brasil.

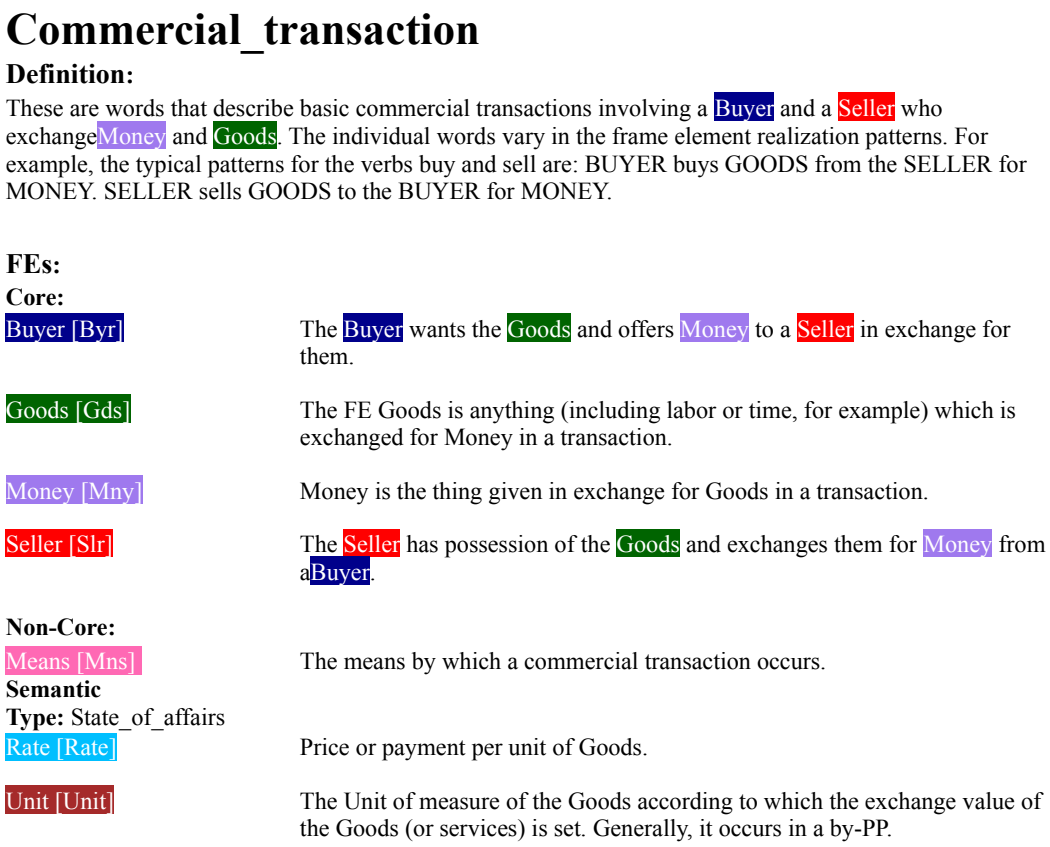
2.3.1 FrameNet

A FrameNet (Baker et al., 1998) é um projeto da Universidade de Berkeley que cria um recurso lexical para o inglês baseada na Semântica de *Frames* de Fillmore e apoiado por evidência extraída de *corpora*. As unidades principais de análise lexical na FrameNet são o ***frame*** e a **unidade lexical**, definida como o par formado por uma palavra com um sentido (a palavra pode ser um verbo, um nome ou um adjetivo). Diz-se que as unidades lexicais **evocam** o *frame* ao qual pertencem.

Os *frames* na FrameNet estão organizados por **domínios**, que são categorias bastante gerais de conhecimento e experiência humanos. As generalizações semânticas através dos *frames* são capturadas mediante a abstração de *frames* gerais e a **herança** destes *frames* por outros mais específicos. Assim, pode-se dizer que cada domínio contém um *frame* geral que captura o que os *frames* mais específicos têm em comum.

Cada *frame* semântico é definido em respeito aos seus ***frame elements***, que são os tipos de entidades que podem participar no *frame* e que podem ser considerados como papéis semânticos mais refinados. Por exemplo, o *frame* `TRANSAÇÃO COMERCIAL` (Figura 2.1),

que caracteriza eventos simples de compra e venda, possui os seguintes *frame elements*: o `COMPRADOR`, o `VENDEDOR`, o `DINHEIRO` e os `BENS`. Diferentes palavras asociadas com este *frame* estão caracterizadas pelos diferentes tipos de sintagma e funções gramaticais que usam para fornecer informação sobre estes *frame elements*.



Commercial_transaction

Definition:

These are words that describe basic commercial transactions involving a Buyer and a Seller who exchangeMoney and Goods. The individual words vary in the frame element realization patterns. For example, the typical patterns for the verbs buy and sell are: BUYER buys GOODS from the SELLER for MONEY. SELLER sells GOODS to the BUYER for MONEY.

FEs:

Core:

Buyer [Byr]	The Buyer wants the Goods and offers Money to a Seller in exchange for them.
Goods [Gds]	The FE Goods is anything (including labor or time, for example) which is exchanged for Money in a transaction.
Money [Mny]	Money is the thing given in exchange for Goods in a transaction.
Seller [Slr]	The Seller has possession of the Goods and exchanges them for Money from aBuyer.

Non-Core:

Means [Mns]	The means by which a commercial transaction occurs.

Semantic Type: State_of_affairs

Rate [Rate]	Price or payment per unit of Goods.
Unit [Unit]	The Unit of measure of the Goods according to which the exchange value of the Goods (or services) is set. Generally, it occurs in a by-PP.

Figura 2.1: Dados do *Frame* `TRANSAÇÃO COMERCIAL`.

Pelos diferentes tipos de dados armazenados na base de dados de FrameNet, é importante caracterizá-la em termos de duas partes:

- **Base de dados Lexical:** contém informação sobre *frames* e *frame elements*, assim como de lemas, lexemas, formas de palavras e categorias gramaticais; i.e., todo o necessário para caracterizar uma unidade lexical.
- **Base de dados de Anotação:** armazena as sentenças anotadas. Para cada palavra alvo sobre a qual a anotação das sentenças exemplo é feita, existe um conjunto de camadas de anotação para os *frame elements*, tipos de sintagma e funções gramaticais.

Atualmente, a FrameNet contém 1.160 *frames* para 12.613 unidades lexicais com

193.862 sentenças anotadas[2]. O *corpus* da FrameNet foi usado na primeira abordagem de aprendizado de máquina estatístico para APS realizada por Gildea e Jurafsky (2002).

FrameNet Brasil

O Projeto FrameNet Brasil (Salomão, 2009) visa construir uma base de dados lexical para o português do Brasil baseado na Semântica de *Frames* e suportado por evidência extraída da combinação de vários *corpora*[3] que representam usos do português do Brasil:

1. ANCIB: *corpus* criado a partir de mensagens enviadas para a lista homônima da Associação Nacional de Pesquisa e Pós-Graduação em Ciência da Informação (até Novembro de 2003) e para a lista abarreto-l, após essa data;

2. ECI-EBR: é uma seleção de excertos de obras brasileiras, contendo pelo menos discurso literário, didático e oral cuidado (discursos políticos);

3. LF (Legendas de Filmes): contém legendas de filmes em Português do Brasil cedidas pelo portal OpenSubtitles.org;

4. NILC/São Carlos: contém textos brasileiros do registro jornalístico (do qual se originou o CETENFolha), didático, epistolar e redações de alunos;

5. NURC-RJ: *corpus* constituído por entrevistas gravadas nas décadas de 1970 e 1990, num total de 350 horas, com informantes de nível superior completo, nascidos no Rio de Janeiro e filhos de pais preferencialmente cariocas.

O *corpus* (3) foi disponibilizado pelo portal OpenSubtitles.org[4], o *corpus* (5), pelo Projeto Norma Linguística Urbana Culta - RJ[5] e os demais estão acessíveis no portal Linguateca[6]. Os *corpora* acima totalizam pouco menos de 72 milhões de palavras. Na Fig. 2.2 apresenta-se, como exemplo, o *frame* `COMÉRCIO_PAGAR`.

2.3.2 VerbNet

A VerbNet (Kipper-Schuler, 2005) é um léxico hierárquico de verbos, independente de domínio e de ampla cobertura, inspirado pelo trabalho de Levin (1993) em classes verbais e suas alternâncias sintáticas.

[2] `https://framenet.icsi.berkeley.edu/fndrupal/current_status`. Acessado em outubro 2012.
[3] Em Salomão (2009) indica-se uma lista maior de *corpora* que compõem a base de dados. Contudo, na página do projeto (`http://www.framenetbr.ufjf.br/`), atualizada em 2011, só aparecem os aqui apresentados.
[4] `http://www.opensubtitles.org/`
[5] `http://www.letras.ufrj.br/nurc-rj/`
[6] `http://www.linguateca.pt/`

Comércio_pagar [Commerce_pay]

Definição

Este frame envolve um Comprador pagando com Dinheiro por Bens. Neste frame, o Dinheiro costuma ser o objeto direto e é mapeado como tema da transferência.

Elementos do Frame

Nucleares (Core)

Comprador [Buyer]	O Comprador tem o Dinheiro e quer os Bens.
Bens [Goods]	O elemento nuclear Bens é qualquer coisa (incluindo trabalho ou tempo, por exemplo) que é trocada por dinheiro em uma transação.
Dinheiro [Money]	O Dinheiro é algo dado na troca pelos Bens em uma transação.
Valor [Rate]	Em alguns casos, o preço ou o pagamento é descrito pela unidade de Bens.
Vendedor [Seller]	O Vendedor tem os Bens e quer o Dinheiro.

Não-nucleares (Non-core)

Razão [Reason]	Razão pela qual um evento ocorre.
Tipo Semântico	
Estado_de_coisas	
Tempo [Time]	Quando o evento ocorre.
Tipo Semântico	
Tempo	
Unidade [Unit]	Este elemento de frame é qualquer unidade em que bens ou serviços podem ser medidos. Geralmente isso ocorre por um sintagma preposicional.
Circunstâncias [Circumstances]	Circunstâncias descrevem o estado de mundo (em um determinado tempo e lugar) que é especificamente independente do evento em si e de qualquer de seus participantes.
Frequência [Frequency]	Este elemento de frame é definido pelo número de vezes que um evento ocorre por alguma unidade de tempo.
Modo[Manner]	Qualquer descrição do evento de pagamento que não é abrangido por elementos nucleares mais específicos, incluindo efeitos secundários (silenciosamente, em voz alta), e descrições gerais comparando eventos (do mesmo modo). Isso também pode indicar categorias marcantes do Comprador que afetam a ação (presunçosamente, friamente, deliberadamente, ansiosamente, cuidadosamente).
Tipo Semântico	
Modo	
Meios [Means]	Os Meios pelos quais uma transação comercial ocorre.
Tipo Semântico	
Estado_de_coisas	
Lugar [Place]	Lugar onde o evento acontece.
Tipo Semântico	
Relação_de_lugar	
Finalidade [Purpose]	A Finalidade para a qual um ato intencional é realizado.
Tipo Semântico	
Estado_de_coisas	

Figura 2.2: Dados do *Frame* COMÉRCIO_PAGAR da FrameNet Brasil.

A VerbNet estende a classificação de Levin de alternâncias sintáticas compartilhadas, tornando explícitas algumas das relações entre sintaxe e semântica. Isto é conseguido através da atribuição de papéis semânticos[7] para cada argumento sintático em uma classe

[7] Kipper-Schuler (2005) emprega o termo **papel temático** mas, para manter consistência na termino-

verbal dada, assim com o uso de predicados semânticos que denotam as relações entre participantes e eventos. Uma classe na VerbNet possui os seguintes componentes[8]:

- **Hierarquia de Classe:** contém a estrutura de árvore de uma classe verbal, incluindo todas as classes pai e subclasses. Cada classe individual é hierárquica no sentido que as classes podem incluir uma ou mais subclasses (Fig. 2.3).

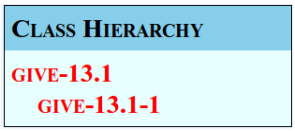


Figura 2.3: Hierarquia da classe give-13.1.

- **Membros:** contém a lista de verbos que pertencem a uma classe ou subclasse específica (Fig. 2.4).

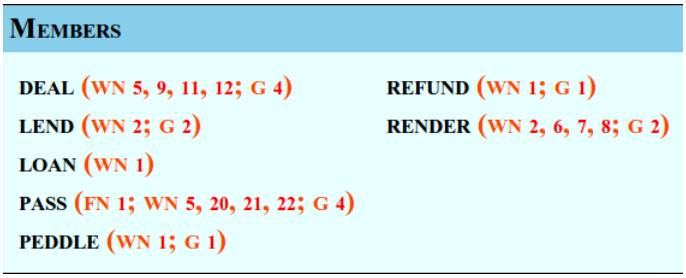


Figura 2.4: Membros da classe give-13.1.

- **Papéis Semânticos:** referem-se a relação semântica entre um predicado e os seus argumentos. Para cada classe, são listados os papéis considerados fundamentais para o comportamento dos membros verbais (Fig. 2.5).

- **Restrições de Seleção:** cada papel semântico listado em uma classe pode ser adicionalmente caracterizado por certas restrições de seleção, que fornecem mais informação sobre a natureza de um determinado papel (Fig. 2.5).

- ***Frames* Sintáticos:** fornecem uma descrição das diferentes realizações superficiais e alternâncias de diátese permitidas para os membros da classe. Esta seção consiste de construções sintáticas, sentenças exemplo e papéis semânticos mapeados aos argumentos sintáticos. Os predicados semânticos também são representados, indicando como os participantes estão envolvidos no evento (Fig. 2.6).

logia usada nesta monografia, decidiu-se usar **papel semântico**.

[8]Os exemplos apresentados correspondem à versão 3.2 da VerbNet.

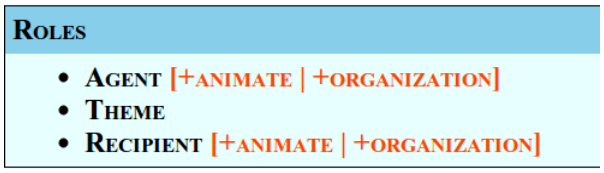


Figura 2.5: Etiquetas de papéis semânticos da classe give-13.1.

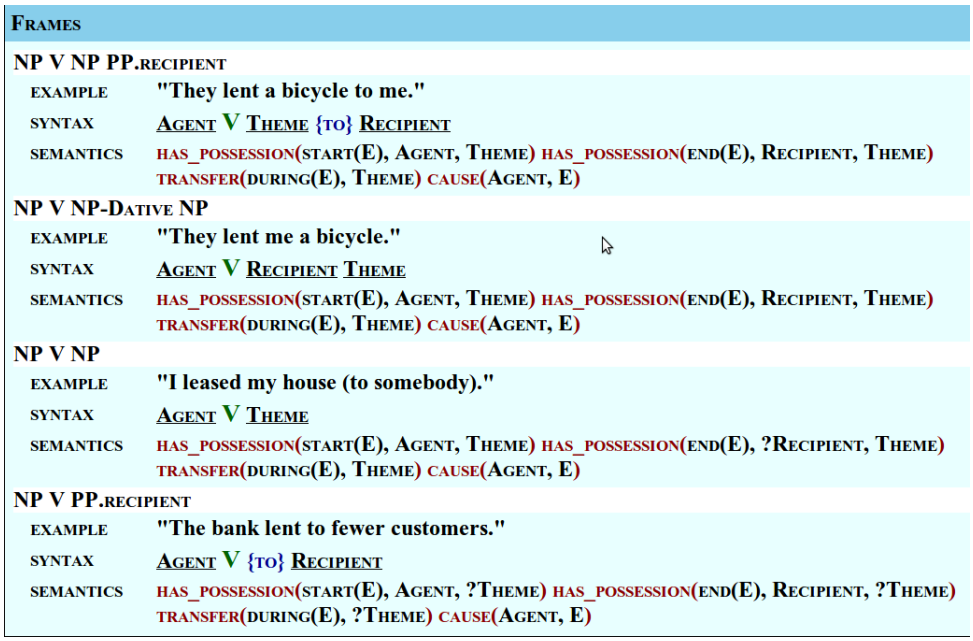


Figura 2.6: *Frames* para a classe give-13.1.

À versão original da VerbNet (Kipper-Schuler, 2005), foram integradas as classes propostas por Korhonen e Briscoe (2004) e Kipper et al. (2006), resultando em um recurso disponível livremente que se constitui na mais compreensível e versátil classificação de verbos para inglês, seguindo o modelo de Levin. Algumas estatísticas extraídas da VerbNet[9] são apresentadas na Tabela 2.2.

VerbNet.Br

O projeto VerbNet.Br (Scarton e Aluísio, 2012) tem como objetivo criar um recurso lexical para o português do Brasil de mesmas características da VerbNet. Com base na hipótese de que as classes de Levin possuem um potencial *cross*-linguístico, é proposto um método semiautomático de 4 etapas que emprega outros recursos lexicais computacionais disponíveis: WordNet (Fellbaum, 1998), WordNet.Br (Dias-da-Silva et al., 2002; Dias-da-Silva, 2004; Dias-da-Silva et al., 2006) e VerbNet. As etapas de construção são:

[9] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html Acessado em outubro 2012.

Tabela 2.2: Resumo da Extensão da VerbNet

	VerbNet Original	VerbNet Estendida
Classes do primeiro nível	191	274
Papéis semânticos	21	23
Predicados semânticos	64	94
Restrições sintáticas	3	55
Número de sentidos de verbos	4.656	5.257
Número de lemas	3.445	3.769

1. **Etapa 1 - Manual:** tradução manual das alternâncias de diátese da VerbNet ao português. Só são consideradas as alternâncias que podem ser diretamente traduzidas. Se alguma alternância não acontece no português ou se acontece em uma forma diferente, não é traduzida.

2. **Etapa 2 - Automática:** busca das alternâncias de diátese dos verbos em *corpus*. Nesta etapa usou-se uma ferramenta para extração de *frames* de subcategorização (Zanette et al., 2012) e os *corpora* PLN-BR-FULL (Muniz et al., 2007), Lácio-Ref (Aluísio et al., 2004) e um *corpus* com textos da Revista Pesquisa FAPESP (Aziz e Specia, 2011). Foram identificados 3.779 lemas de verbos (com frequência superior a dez ocorrências), 408 *frames* sintáticos sem parametrização por preposição e 3.578 *frames* sintáticos com parametrização (descartando aqueles com frequência inferior a cinco ocorrências).

3. **Etapa 3 - Automática:** geração de candidatos a membros das classes da VerbNet.Br aproveitando os mapeamentos VerbNet – WordNet e WordNet – WordNet.Br. Foram identificados 4.298 lemas de verbos para 254 classes, com uma média de 16 verbos por classe (aqui foram trazidas informações para todas as 274 classes da VerbNet – na etapa de validação (Etapa 4) é que foram consideradas apenas 213 classes). Das 213 classes consideradas para a primeira versão da VerbNet.Br, 10 não apresentaram alinhamentos com a WordNet.Br e por isso foram descartadas.

4. **Etapa 4 - Automática:** escolha automática dos membros das classes da VerbNet.Br. Para cada candidato a membro (definidos na Etapa 3) buscou-se os respectivos *frames* sintáticos correspondentes ao verbo candidato (alternâncias encontradas na Etapa 2). Compararam-se os *frames* sintáticos do candidato com os definidos para a classe (definidas na Etapa 1) a qual ele é candidato a membro. Se o verbo possuía pelo menos o teto de 10% dos *frames* sintáticos definidos para a classe ele se tornava membro dela. Caso contrário, o candidato foi marcado como não membro.

Os papéis semânticos, as restrições de seleção e os predicados semânticos são diretamente herdados da VerbNet. Embora o método usado seja *cross*-linguístico (explora as compatibilidades entre o inglês e o português), uma revisão linguística dos resultados obtidos pelo método semiautomático é altamente desejável.

2.3.3 *PropBank*

O projeto PropBank (Palmer et al., 2005) adiciona informação predicado–argumento, ou papéis semânticos, às estruturas sintáticas do Penn Treebank[10] (Marcus et al., 1993). Define-se um conjunto de papéis semânticos subjacentes para cada verbo, assim como papéis tradicionalmente vistos como argumentos e adjuntos, e anota-se cada instância no texto do Penn Treebank II[11]. Um dos objetivos é fornecer um *corpus* anotado que possa ser usado no treinamento de sistemas de aprendizado de máquina.

Devido à dificuldade de definir um conjunto universal de papéis semânticos que abranja todos os tipos de predicado, o PropBank define papéis semânticos para cada verbo. Os argumentos semânticos de um verbo em particular estão numerados, começando com zero. Para um verbo em particular, `ARG0` é geralmente o argumento que exibe os atributos de um Agente Prototípico (Dowty, 1991), enquanto `ARG1` é um Paciente Prototípico ou Tema. Não se podem fazer generalizações entre verbos para os argumentos de números maiores, apesar de que foi feito um esforço para definir, consistentemente, papéis através dos membros das classes da VerbNet. Na Fig. 2.7, apresentam-se os papéis específicos numerados do verbo *break* no seu primeiro sentido.

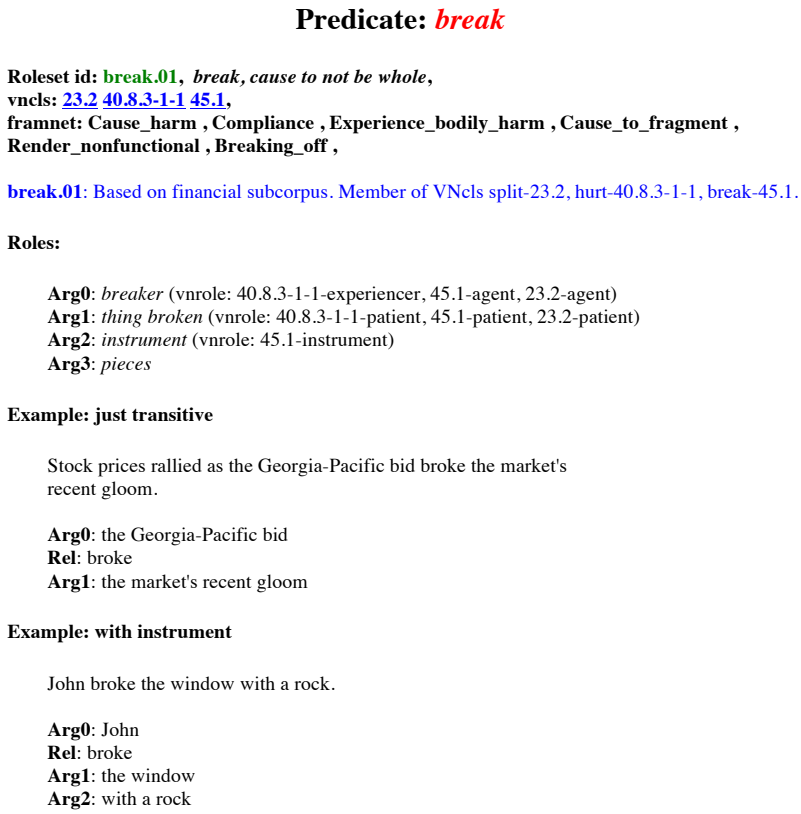
Um conjunto de papéis que corresponde a um uso distintivo de um verbo é chamado de ***roleset*** e pode ser associado com um conjunto de *frames* sintáticos que indicam as variações sintáticas permitidas na expressão desse conjunto de papéis. O *roleset* com seus *frames* associados é chamado de ***frameset***. Um verbo polissêmico poderia ter mais de um *frameset* quando as diferenças em significado são suficientemente distintas para justificar um conjunto de papéis diferente; um para cada *frameset*.

Cada papel semântico possui um campo descritor, mas que é usado como documentação durante a anotação e não tem nenhum suporte teórico. Além disso, cada *frameset* é complementado por um conjunto de exemplos, que tentam cobrir o escopo de alternâncias sintáticas permitidas por esse uso. A coleção de entradas do *frameset* para um verbo é chamada de ***frame file*** do verbo.

Na versão atual do PropBank, apresenta-se um mapeamento entre os *rolesets* deste com as classes da VerbNet e os *frames* da FrameNet. Isto como resultado do projeto

[10]Um Treebank é um *f* cujas sentenças já possuem algum tipo de anotação, neste caso anotação sintática.

[11]O Penn Treebank II contém 1 milhão de palavras do *Wall Street Journal* de 1989.



Predicate: *break*

Roleset id: break.01, *break, cause to not be whole*,
vncls: 23.2 40.8.3-1-1 45.1,
framnet: Cause_harm , Compliance , Experience_bodily_harm , Cause_to_fragment , Render_nonfunctional , Breaking_off ,

break.01: Based on financial subcorpus. Member of VNcls split-23.2, hurt-40.8.3-1-1, break-45.1.

Roles:

Arg0: *breaker* (vnrole: 40.8.3-1-1-experiencer, 45.1-agent, 23.2-agent)
Arg1: *thing broken* (vnrole: 40.8.3-1-1-patient, 45.1-patient, 23.2-patient)
Arg2: *instrument* (vnrole: 45.1-instrument)
Arg3: *pieces*

Example: just transitive

Stock prices rallied as the Georgia-Pacific bid broke the market's recent gloom.

Arg0: the Georgia-Pacific bid
Rel: broke
Arg1: the market's recent gloom

Example: with instrument

John broke the window with a rock.

Arg0: John
Rel: broke
Arg1: the window
Arg2: with a rock

Figura 2.7: Entrada no PropBank para o verbo *break.01*

SemLink[12] (Loper et al., 2007) que visa ligar estes recursos lexicais usando um conjunto de mapeamentos, permitindo combinar as diferentes informações fornecidas por eles. Um dos benefícios imediatos desse tipo de mapeamento é a capacidade de agrupar automaticamente as descrições de argumentos do PropBank, os papéis semânticos da VerbNet e os *frame elements* da FrameNet, em etiquetas de argumentos específicas do PropBank (como apresentado na Fig. 2.7).

Embora a maioria de *rolesets* tenha de dois a quatro papéis numerados, alguns podem ter até seis, em particular para alguns verbos de movimento. Não se realiza nenhuma distinção entre argumentos e adjuntos. Embora muitos linguistas possam considerar qualquer argumento acima de `ARG2` ou `ARG3` como adjunto, alguns aparecem com tanta frequência com os seus respectivos verbos, ou classes de verbos, que são atribuídos números para poder assegurar consistência na anotação.

Além destes papéis numerados específicos para cada verbo, o PropBank define vários outros papéis que são mais gerais e que podem ser aplicados para qualquer verbo chamados de `ARGMs` (ver Tabela 2.3). Apesar de não ser considerados adjuntos, `NEG` para negação

[12] http://verbs.colorado.edu/semlink/

no nível verbal e MOD para verbos modais também são incluídos nesta lista para permitir que todo constituinte em torno do verbo seja anotado.

Tabela 2.3: Tipos de etiqueta ARGM.

Etiqueta	Descrição
LOC	local
EXT	extensão
DIS	conectivos discursivos
ADV	propósito geral
NEG	marcador de negação
MOD	verbo modal
CAU	causa
TMP	tempo
PNC	propósito
MNR	maneira
DIR	direção

A anotação dos sintagmas preposicionais das sentenças possui um tratamento especial. Por exemplo, na sentença:

João colocou o vaso na mesa. (2.5)

se fosse definido um papel destino, este seria claramente atribuído a a mesa, e o sintagma nominal que é núcleo do sintagma preposicional seria anotado como o argumento. Mas por outro lado, ARGMs que são sintagmas preposicionais são anotados no nível deste sintagma e não no seu núcleo. Assim, para ter uma anotação consistente, os argumentos numerados também são anotados no nível do sintagma preposicional.

Originalmente, o PropBank só continha anotação para papéis semânticos de verbos, mas agora também inclui para substantivos e adjetivos. O estado atual deste recurso pode ser visto na Tabela 2.4[13].

Tabela 2.4: Estado atual da anotação do PropBank.

	Frame Files	Predicados	*Framesets*
Verbos	5.652	6.379	7.648
Substantivos	1.405	1.472	1.778
Adjetivos	85	85	90

[13] http://verbs.colorado.edu/propbank/propbank-status-en.html Acessado em outubro 2012.

PropBank.Br

O projeto PropBank.Br (Duran e Aluísio, 2012) visa, em primeiro lugar, a anotação de um *Treebank* de português do Brasil com papéis semânticos seguindo as diretrizes do projeto PropBank. O *corpus* resultante está sendo usado na construção de um léxico de predicados verbais do português e suas estruturas predicado–argumento.

Como no projeto PropBank, um dos principais objetivos é acrescentar uma camada de anotação semântica a um *corpus* anotado sintaticamente e manualmente corrigido. Foi selecionado o *corpus* Bosque da Floresta Sintá(c)tica[14], anotado pelo *parser* Palavras (Bick, 2000) e revisado manualmente por linguistas. Na sua versão 8.0, este *corpus* está composto por 9.437 árvores sintáticas revistas, correspondendo a 1.962 extratos, 215.420 unidades, aproximadamente 183.619 palavras, retiradas dos *corpus* CETENFolha e CETENPúblico (Santos et al., 2007).

Na anotação predicado–argumento só foram considerados os verbos principais das proposições, descartando os auxiliares (temporais, modais e aspectuais), que receberão uma anotação apropriada no futuro. Se uma sentença tem mais de um verbo alvo de anotação, i.e., que possui uma estrutura argumental, então essa sentença é repetida para cada verbo alvo de anotação. Assim, as 4.213 sentenças da seção CETENFolha (relativa à variante do português do Brasil do Bosque) produziram 6.142 instâncias para anotação com 1.068 verbos alvo. Na Fig. 2.8 apresenta-se uma sentença anotada do *corpus* PropBank.Br para o verbo abrir.

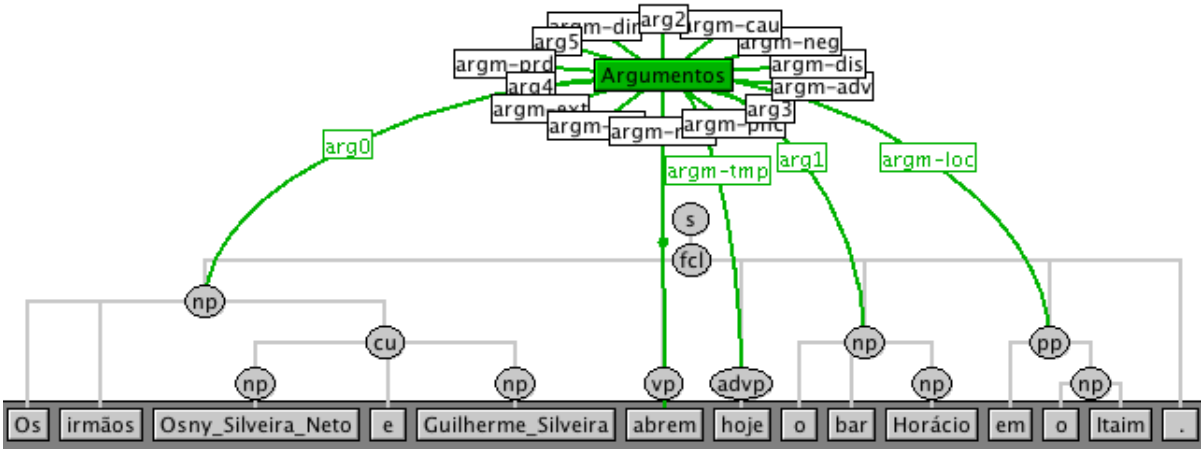


Figura 2.8: Sentença anotada no PropBank.Br para o verbo abrir visualizada com a ferramenta SALTO (Burchardt et al., 2006).

[14]http://www.linguateca.pt/Floresta/principal.html

CINTIL – PropBank

Construído com base no CINTIL–DeepGramBank (Branco et al., 2010), que é um *corpus* anotado com representações linguísticas profundas para o português, O CINTIL–PropBank (Branco et al., 2012) é um *corpus* de sentenças anotadas com a suas estruturas de constituintes e etiquetas de papéis semânticos, composto de 10.039 sentenças e 110.166 *tokens* obtidos de diferentes recursos e gêneros: notícias (8.861 sentenças; 101.430 *tokens*) e novelas (399 sentenças; 3.082 *tokens*). Além disso, possui 779 sentenças (5.654 *tokens*) usadas para provas de regressão da gramática computacional que suporta a anotação do *corpus*: LXGram (Branco e Costa, 2010).

Para criar este PropBank, adotou-se uma análise semi-automática com anotação *double-blind* seguida de julgamento. O conjunto de dados resultante contém três níveis de informação: sintagmas constituintes, funções gramaticais e papéis semânticos dos sintagmas. A principal motivação para a criação deste recurso foi construir um conjunto dados de alta qualidade com informação semântica que possa suportar o desenvolvimento de anotadores de papéis semânticos para o português.

2.4 Considerações Finais

Neste capítulo foi apresentada a noção de papel semântico e como ela foi desenvolvida nas teorias de semântica lexical de Charles Fillmore, desde os papéis de caso até os *frames* semânticos. Também foi apresentado o conceito de alternância de diátese e sua importância na construção das classes verbais de Beth Levin. A alternância sintática faz da anotação de papéis semânticos uma tarefa tanto desafiadora quanto útil.

Foram descritos os principais recursos lexicais disponíveis – FrameNet, VerbNet e PropBank – que fornecem bases de dados anotados que podem ser utilizados no desenvolvimento de diferentes sistemas de PLN. De interesse particular é o PropBank que tem como propósito fornecer dados de treinamento (e teste) para a implementação de sistemas automáticos de APS baseados em técnicas de aprendizado de máquina.

Como o objetivo deste mestrado é desenvolver sistemas automáticos de APS para o português do Brasil, os dados do *corpus* PropBank.Br (versão para o português do Brasil do PropBank) são usados na sua implementação. Assim, ao menos que seja especificado o contrário, assumi-se o estilo de anotação do projeto PropBank no restante desta monografia.

Capítulo

3

Anotação Automática de Papéis Semânticos

A anotação de papéis semânticos (APS) permite analisar parte do significado de uma sentença através da informação fornecida pelas relações entre o verbo e os constituintes da sentença. Uma variedade de aplicações do processamento de língua natural (PLN) – como extração de informação (Surdeanu et al., 2003; Moreda et al., 2007), sistemas de perguntas & respostas (Stenchikova et al., 2006; Frank et al., 2007; Shen e Lapata, 2007; Stoyanchev et al., 2008), sumarização (Melli et al., 2005; Suanmali et al., 2010) e tradução automática (Wu e Fung, 2009a,b) – podem se beneficiar desta capacidade na análise das entradas ingressadas pelos usuários (Màrquez, 2009). Este capítulo apresenta meios de realização desta análise semântica de forma automática.

Os sistemas de APS automática podem ser **baseados em *corpus*** e usar *corpus* anotados previamente construídos, ou **baseados em conhecimento** e usar conhecimento linguístico previamente adquirido (Moreda Pozo, 2008). Considerando o objetivo do projeto, só os primeiros serão detalhados neste capítulo[1].

Inicialmente (Seção 3.1), explicam-se alguns conceitos básicos da área de Aprendizado de Máquina (AM); abordagem usada pelos sistemas baseados em *corpus* para predizer os papéis semânticos dos argumentos dos verbos de sentenças. Após, descreve-se o funcionamento básico de um sistema baseado em *corpus* (Seção 3.2) e alguns dos sistemas mais representativos desta abordagem (Seção 3.3). Depois disso, revisam-se trabalhos para o português em APS automática e tarefas relacionadas (Seção 3.4), terminando com algumas considerações finais (Seção 3.5).

[1]Consultar Moreda Pozo (2008) para referências sobre a segunda abordagem.

3.1 Conceitos Básicos de Aprendizado de Máquina

Um programa de computador **aprende** a partir de uma experiência E com respeito a alguma classe de tarefas T e medida de desempenho P, se o seu desempenho em tarefas de T, medido por P, melhora com a experiência E em relação a um *baseline* (Mitchell, 1997). Assim, um sistema de aprendizado tem a função de analisar as informações de E e generalizá-las, para a extração de novos conhecimentos (Monard e Baranauskas, 2003).

A experiência E atua como entrada do sistema de aprendizado e recebe o nome de **conjunto de treinamento**, formado por uma coleção de **instâncias** (objetos específicos de E), cada uma representada por um **vetor de atributos**. Como indicado em Zhu e Goldberg (2009), a predição desejada em uma instância recebe o nome de **etiqueta**, e esta pode vir de um conjunto finito de valores chamados de **classes** (Zhu e Goldberg, 2009). A dificuldade fundamental do aprendizado de máquina estatístico é generalizar a predição a partir de um conjunto finito de treinamento para **dados de teste** não vistos.

O aprendizado é considerado **supervisionado** se o conjunto de treinamento consiste de pares instância–etiqueta chamados de **dados anotados**. Dependendo se as classes são discretas ou contínuas, o problema é chamado de **classificação** ou de **regressão**, respectivamente. Se as instâncias do conjunto de treinamento não contêm etiquetas que supervisionem o aprendizado, este é chamado de **não supervisionado**. Uma tarefa comum deste tipo de aprendizado é ***clustering***, no qual as instâncias são separadas em agrupamentos (*clusters*) de acordo com alguma medida de similaridade.

O aprendizado **semissupervisionado** emprega estratégias que estendem o aprendizado supervisionado ou o não supervisionado para incluir informação adicional do outro paradigma de aprendizado. Por exemplo, a **classificação semissupervisionada** tem como objetivo treinar um classificador com dados anotados e não anotados, para obter um melhor classificador do que se fosse treinado só com dados anotados. Tipicamente, assume-se que existem muito mais dados não anotados do que anotados.

Existem muitas tarefas para as quais há uma grande escassez de dados anotados e que pode ser difícil de se obter as etiquetas (por exemplo, porque há a necessidade de anotadores humanos), mas dados não anotados podem ser abundantes e fáceis de coletar. Por isso, o aprendizado semissupervisionado é atrativo, porque pode, potencialmente, usar dados tanto anotados quanto não anotados para atingir um desempenho melhor do que o aprendizado supervisionado. Com uma perspectiva diferente, o aprendizado semissupervisionado pode atingir o mesmo nível de desempenho do que o aprendizado supervisionado, mas com menos instâncias anotadas, o que reduz o esforço na anotação.

3.2 Funcionamento Básico de um Sistema Baseado em *corpus*

De forma geral, trata-se a APS como um problema de classificação, no qual deve-se predizer uma etiqueta de papel semântico para cada palavra (ou conjunto de palavras) de uma sentença, considerando a sua relação com o verbo. Convém dividir o funcionamento completo de um sistema de APS automática em três grandes fases:

1. **Identificação do Verbo Alvo:** consiste em determinar o verbo (ou verbos) na sentença que possui uma estrutura argumental que deve ser anotada. Às vezes, esta informação é fornecida pelo usuário e não precisa ser identificada automaticamente.

   João [quebrou $_{\mathsf{V}}$] a janela com a pedra. (4.1)

2. **Identificação de Argumentos:** consiste em dividir a sentença em conjuntos de palavras que são candidatos a argumentos do verbo. Geralmente, extraem-se atributos da árvore sintática da sentença para identificar quais grupos de palavras não podem ser divididos e, portanto, não podem ser candidatos individuais a argumentos (por exemplo, itens lexicais que correspondem a constituintes diferentes da árvore).

   [João $_{\mathsf{ARG}}$] [quebrou $_{\mathsf{V}}$] [a janela $_{\mathsf{ARG}}$] [com a pedra $_{\mathsf{ARG}}$]. (4.2)

3. **Classificação de Argumentos:** consiste em extrair atributos dos candidatos a argumentos, identificados no passo anterior, para determinar qual etiqueta de papel semântico correspondente será atribuída a cada um deles. A maioria das vezes, a árvore sintática fornece a informação necessária para realizar esta classificação, mas também outros recursos lexicais (*VerbNets*, *WordNets*, reconhecedores de entidades nomeadas[2], etc.) podem ser aproveitados.

   [João $_{\mathsf{ARG0}}$] [quebrou $_{\mathsf{V}}$] [a janela $_{\mathsf{ARG1}}$] [com a pedra $_{\mathsf{ARG2}}$]. (4.3)

Para **avaliar** a anotação automática, usam-se, tradicionalmente, as medidas de precisão (porcentagem das etiquetas colocadas pelo sistema que estão certas), cobertura (porcentagem de etiquetas certas, corretamente identificadas pelo sistema), acurácia (porcentagem de etiquetas certas, quando os limites de todos os argumentos são corretos) e F_1 (média harmônica da precisão e da cobertura).

[2]Refere-se à detecção dos significados (ou categorias ontológicas) de nomes próprios.

3.3 Sistemas Desenvolvidos

Nesta seção, descrevem-se alguns sistemas representativos da APS automática baseada em *corpus*, classificados segundo a abordagem de aprendizado de máquina empregada para predizer as etiquetas semânticas. Para cada um deles, especifica-se o **conjunto de papéis semânticos** empregado na anotação, o ***corpus*** de sentenças usadas como dados de treinamento, o **algoritmo** de aprendizado, a **estratégia** de anotação seguida, e os **atributos** (lexicais, sintáticos e/ou semânticos) dos constituintes das sentenças do *corpus* empregados pelo algoritmo.

3.3.1 Sistemas Supervisionados

a) O sistema de Gildea e Jurafsky (2002) foi o primeiro em empregar técnicas estatísticas para extrair informação de um grande *corpus* de sentenças anotadas e utilizá-la para APS automática. O ***corpus*** de treinamento é a FrameNet e, consequentemente, os seus *frame elements* constituem o **conjunto de papéis semânticos**. Como **estratégia** de anotação, indentifica-se manualmente o predicado[3] que evoca o *frame* e o próprio *frame*, para depois identificar e classificar os argumentos automaticamente. Uma das contribuições mais importantes deste sistema é o conjunto de **atributos** empregado, porque é utilizado (com certas extensões) por quase a totalidade de sistemas de APS desenvolvidos posteriormente. A partir da árvore sintática automática da sentença obtida usando o *parser* de Collins (1999), extraem-se os seguintes atributos:

- **Tipo de Sintagma:** indica a categoria sintática do sintagma que expressa o papel semântico. Dada a árvore sintática, encontra-se o constituinte que abrange o mesmo conjunto de palavras que cada *frame element* anotado, e a etiqueta não terminal do constituinte é tomada como o tipo de sintagma.
- **Categoria Principal:** indica se um determinado sintagma nominal (`NP`) é sujeito ou objeto direto do verbo. De acordo com a anotação sintática do Penn Treebank, nós `NP` que se encontram embaixo de nós `S` são geralmente sujeitos gramaticais, e nós `NP` embaixo de nós `VP` são geralmente objetos. Assim, sobe-se pela árvore sintática desde o constituinte que corresponde a um *frame element* até encontrar um nó `S` ou `VP`, o que determina o valor deste atributo.
- **Caminho na Árvore Sintática:** define o caminho na árvore sintática desde o predicado evocador do *frame* semântico até o constituinte a ser anotado. É representado como uma cadeia de nós não terminais da árvore sintática, unidos por

[3] O **predicado evocador** é também chamado de **predicado alvo**. Como este sistema foi desenvolvido usando a FrameNet, não se limita a predicados verbais.

símbolos que indicam movimentos ascendentes ou descendentes através da árvore. O primeiro elemento da cadeia é a função gramatical do predicado evocador e o último é o tipo de sintagma ou categoria sintática do constituinte da sentença marcado como *frame element*.

- **Posição:** indica se o constituinte que será anotado aparece antes ou depois do predicador evocador do *frame* semântico. Este atributo possui correlação com a função gramatical, uma vez que os sujeitos geralmente aparecerão antes de um predicado verbal e os objetos depois.
- **Voz:** indica se o verbo está na voz ativa ou passiva, usando um conjunto de 10 padrões (elaborados pelos autores) de identificação de verbos na voz passiva. Cada padrão requer tanto um auxilar passivo (alguma forma de *to be* ou *to get*) e um verbo em particípio passado.
- **Núcleo do Sintagma:** indica o núcleo do sintagma do constituinte a ser anotado. No caso dos sintagmas nominais, fornece informação adicional que pode ser usada como restrições de seleção. Vale mencionar que, no caso dos sintagmas preposicionais, o núcleo é a preposição.
- **Subcategorização:** indica a regra da estrutura do sintagma que expande o nó pai do predicado evocador na árvore sintática, para diferenciar usos transitivos e intransitivos do verbo. Este atributo só é usado para predicados verbais.

Adicionalmente, foi usado o atributo **Conjunto de Papéis**, que indica todos os papéis que podem ser atribuídos por um determinado predicado em uma sentença. Este atributo é extraído do *frame* ao qual o predicado alvo pertence.

O **algoritmo** de aprendizado combina probabilidades de distribuições condicionadas sobre uma variedade de subconjuntos dos atributos. Devido à esparsidade dos dados, não é possível estimar a probabilidade condicionada de cada papel dados os atributos descritos. Assim, calculam-se as probabilidades para cada subconjunto dos atributos e interpolam-se como uma combinação linear das distribuições resultantes. Esta interpolação é realizada sobre as distribuições mais específicas para as quais existem dados disponíveis.

Para os experimentos, 10% das sentenças anotadas para cada predicado alvo foram reservadas para teste e outro 10% para desenvolvimento. Aquelas (poucas) palavras alvo que tinham menos de 10 sentenças anotadas foram removidas do *corpus*. Assim, o número médio de sentenças para cada palavra alvo é 34, e o número de sentenças por *frame* é 732. Os **resultados** obtidos indicam uma acurácia de 82% na classificação de papéis semânticos (os argumentos foram identificados previamente), e $64,6\%$ de precisão e 64% de cobertura na tarefa combinada (identificação+classificação).

Este mesmo sistema foi utilizado por Palmer et al. (2005) em uma versão preliminar do ***corpus*** PropBank. Para poder oferecer resultados comparáveis aos obtidos com a FrameNet, as sentenças do *corpus* foram analisadas também com o *parser* Collins e foram excluídos os predicados com menos de 10 exemplos. Os **resultados** obtidos utilizando informação da análise sintática automática foram 79,9% de acurácia na classificação de papéis; e 68,6% de precisão e 57,8% de cobertura no caso da tarefa combinada. Usando a informação da análise sintática manual, obteve-se 82% de acurácia na classificação de papéis e 74,3% de precisão e 66.4% cobertura na tarefa combinada.

b) Pradhan et al. (2008) analisam a robustez de um sistema de APS automática quando treinado com dados de um gênero e testado em outro. A **estratégia** usada consiste em treinar um classificador multi-classe com *Support Vector Machines* (SVMs) como **algoritmo** de aprendizado. Usa-se a abordagem *One vs All* para treinar um classificador para cada etiqueta de papel semântico existente no *corpus*. Os atributos usados pelo sistema são:

- **Verbo:** a forma e o lema do verbo cujos argumentos são identificados.
- **Caminho**, **Tipo de Sintagma**, **Posição**, **Voz**, **Subcategorização** e **Núcleo do Sintagma:** como definidos por Gildea e Jurafsky (2002).
- ***Cluster* do Verbo:** indica a classe do verbo alvo dentro das 64 criadas usando o modelo de co-ocorrência de Hofmann e Puzicha (1998) e a base de dados de relações verbo–objeto direto de Lin (1998).
- **POS do Núcleo:** *part-of-speech* do **Núcleo do Sintagma**.
- **Entidade Nomeada no Constituinte:** atributos binários para 7 entidades nomeadas anotadas automaticamente.
- **Generalizações do Caminho**: quatro variações do atributo **Caminho**. Por exemplo, **caminho parcial**, indica o caminho na árvore sintática desde o constituinte até o menor antepassado comum do verbo e o constituinte. Os outros atributos são: **caminho de frases** (com 4 variações), **caminho de n-gramas** e **caminho de tipo de sintagma de um caractere**.
- **Contexto do verbo:** duas palavras antes e duas depois do verbo, assim como suas etiquetas de *part-of-speech*.
- **Pontuação:** sinais de pontuação à esquerda e à direita do constituinte.
- **Núcleo do Sintagma Preposicional:** se o constituinte é um sintagma preposicional, considerar o núcleo do primeiro sintagma nominal dentro dele.

- **Primeira e Última Palavra/POS no Constituinte:** a primeira e última palavra no constituinte junto com sua *part-of-speech*.
- **Posição Ordinal do Constituinte:** concatenação do tipo de sintagma do constituinte e a posição ordinal dele com respeito ao verbo alvo.
- **Distância em Constituintes na Árvore:** indica o número de constituintes encontrados no **Caminho** desde o verbo até o constituinte a ser anotado.
- **Atributos dos Parentes do Constituinte:** nove atributos que indicam o **Tipo de Sintagma**, **Núcleo** e **Núcleo-POS** para o pai, irmão esquerdo e irmão direito do constituinte a ser anotado.
- **Palavras Temporais:** atributos binários que indicam a presença de um conjunto de palavras temporais que não são anotadas pelo reconhecedor de entidades nomeadas.
- ***Frame* Sintático:** proposto originalmente por Xue e Palmer (2004), é uma modificação do **Caminho** no qual os `NPs` e o verbo são considerados como "pivôs", e os outros constituintes são definidos em relação com eles.

Experimentos no PropBank (***corpus*** baseado no Wall Street Journal (WSJ)) usando árvores sintáticas geradas pelo *parser* de Charniak e Johnson (2005) obtêm 87,8% de precisão, 84,1% de cobertura e 85,9% de F_1 para identificação; 92% de acurácia para classificação; e 81,7% de precisão, 78,4% de cobertura e 80% de F_1 para a tarefa combinada. Experimentos no *corpus* Brown anotado no estilo do PropBank, obtêm 81.2% de F_1 para identificação e 63.9% de F_1 para a tarefa combinada. Os resultados indicam que, enquanto o desempenho na identificação de argumentos é relativamente similar nos dois *corpora*, isso não acontece com a classificação de argumentos. Um dos possíveis motivos é que a maioria dos atributos na etapa de classificação são lexicais/semânticos, enquanto os atributos mais estruturais estão mais presentes na etapa de identificação. Embora não indicados aqui, os autores apresentam mais experimentos, analisando o efeito de árvores sintáticas *gold* e automáticas, mudando o *corpus* de treinamento – só o WSJ, só o Brown, WSJ+Brown, mudando o tamanho do *corpus* de treinamento/teste, entre outros.

c) Punyakanok et al. (2008) comparam o uso de anotação sintática completa (como no Penn Treebank) com superficial (só *chunks*[4] e orações) para a tarefa de APS. Apresentam um sistema que combina uma técnica de AM com um processo de inferência baseado em programação linear inteira que incorpora restrições linguísticas e estruturais em um processo de decisão global. A **estratégia** tem quatro fases:

[4]Um *chunk* é um sintagma que contém palavras relacionadas sintaticamente. Aproximadamente, são obtidos "achatando" uma árvore sintática completa.

1. **Poda:** quando a árvore sintática completa está disponível, todo constituinte é um candidato. Então, usa-se o algoritmo de Xue e Palmer (2004) para eliminar aqueles mais improváveis de realmente serem argumentos.

2. **Identificação:** no caso da árvore sintática completa, usa-se um classificador binário `ARG - NO ARG` e os seguintes **atributos**:

   - **Verbo e POS do Verbo:** indica o lema e a *part-of-speech* do verbo alvo.
   - **Voz**, **Tipo de Sintagma**, **Núcleo**, **POS do Núcleo**, **Posição**, **Caminho** e **Subcategorização:** como definidos por Gildea e Jurafsky (2002).
   - **Contexto:** como definido por Pradhan et al. (2008).
   - **Classe Verbal:** classe na VerbNet do verbo alvo.
   - **Comprimento:** número de palavras e de *chunks* do constituinte a ser anotado.
   - ***Chunk*:** indica se o constituinte a ser anotado `é`, `incorpora`, `sobrepõe` ou `está embutido em` um *chunk*.
   - **Padrão de *Chunks*:** sequência de *chunks* desde o constituinte até o verbo alvo.
   - **Comprimento do Padrão de Chunks:** número de *chunks* no argumento.
   - **Posição Relativa na Oração:** indica a posição do constituinte relativa ao verbo alvo: `irmãos`, `pai do constituinte é atepassado do verbo`, `pai do verbo é antepassado do constituinte` ou `outro`.
   - **Cobertura da Oração:** indica quanto da oração do verbo alvo é coberta pelo constituinte a ser anotado.
   - **NEG:** indica se existe um indicador de negação no *chunk* do verbo alvo.
   - **MOD:** indica se existe um verbo modal no *chunk* do verbo alvo.

   No caso da análise sintática superficial, usam-se dois classificadores binários para indicar o início e o fim do argumento e os seguintes **atributos**:

   - **Tipo de Sintagma:** indica `NP`, `VP` ou `PP` usando uma heurística simples.
   - **Núcleo e POS do Núcleo:** são a palavra mais à direita para `NP` e a mais à esquerda para `VP` e `PP`.
   - **Caminho Superficial:** caminho na pseudo árvore sintática entre o constituinte e o verbo alvo.
   - **Subcategorização Superficial:** a estrutura de *chunks* e orações em torno do pai do verbo alvo na pseudo árvore sintática.

3. **Classificação:** usa-se um classificador multi-classe e o mesmo conjunto de atributos da etapa anterior. No caso da árvore sintática completa, é também usado o

atributo ***frame* sintático** como definido por Xue e Palmer (2004). O **algoritmo** de aprendizado usado tanto nesta etapa como na anterior é uma variação da regra de atualização *Winnow* incorporada em SNoW (Carlson et al., 1999).

4. **Inferência:** tenta incorporar informação global transversal aos argumentos na anotação final. Para isso, inclui-se conhecimento estrutural e linguístico do tipo argumentos não se sobrepõem ou cada verbo tem, no máximo, um argumento de cada tipo na forma de restrições usando programação linear inteira.

Usando o PropBank como ***corpus*** de treinamento e teste, realizaram-se experimentos para avaliar a importância da análise sintática, observando os efeitos de usar uma análise sintática completa ou parcial em cada etapa da APS. No caso da classificação de argumentos, quando os seus limites são conhecidos, a acurácia com análise sintática completa ou superficial é quase igual: 91,5% vs 90,75% para árvores *gold* e 90,32% vs 89,71% para árvores automáticas obtidas com o *parser* de Charniak e Johnson (2005). Para identificação de argumentos, a análise sintática completa permite obter melhores resultados para árvores *gold*: 86,82% vs 84.72% de F_1. Contudo, quando são usadas árvores automáticas, a análise completa não necessariamente permite obter um melhor desempenho global: 84,63% vs 85,08% de F_1. Os autores apresentam mais experimentos realizando uma análise similar para as etapas de poda e inferência; e também mostram como combinar diferentes análises sintáticas para obter um melhor resultado global.

d) Toutanova et al. (2008) apresentam um modelo para APS que efetivamente captura a intuição semântica que o conjunto de argumentos semânticos (principais) de um determinado verbo é uma estrutura conjunta, com fortes dependências entre os argumentos. Modelam-se dependências entre as etiquetas dos constituintes e entre cada etiqueta e os atributos de entrada dos outros constituintes. O **algoritmo** de atribuição conjunta emprega uma abordagem de *re-ranking* (Collins e Koo, 2005) logarítmico linear que seleciona as n atribuições conjuntas sem sobreposição de etiquetas mais prováveis de acordo com um modelo local[5]. A **estratégia** é de duas fases – identificação e classificação, usando modelos logarítmicos lineares em ambas com os seguintes **atributos**:

- **Tipo de Sintagma**, **Lema do Verbo**, **Caminho**, **Posição**, **Voz**, **Núcleo**, **Subcategorização**, **Primeira e Última Palavra do Constituinte**, **Atributos do Parentes do Verbo**, **Caminho Parcial** e **Núcleo do Sintagma Preposicional:** como definidos anteriormente.

[5]Um classificador é local se atribui uma probabilidade a uma etiqueta para um constituinte independentemente das etiquetas dos outros.

- **Núcleo do Pai PP:** Se o pai do constituinte a ser anotado é um sintagma preposicional (`PP`); seu núcleo.
- Combinações de atributos: **Lema do Verbo + Caminho**, **Lema do Verbo + Núcleo**, **Lema do Verbo + Tipo de Sintagma**, **Voz + Posição** e **Lema do Verbo + Núcleo do Pai PP**.
- **Sujeito Ausente:** indica se não existe um sujeito para o verbo alvo. Considera-se que um verbo não tem sujeito se o maior `VP` na cadeia de `VP`s que dominam o verbo não tem um `NP` ou `S(BAR)` como seu irmão esquerdo (considerando a anotação do Penn Treebank).
- **Caminho Projetado:** indica o **Caminho** desde o maior `VP` na cadeia de `VP`s do verbo alvo, até o constituinte a ser anotado.

Foi usado o PropBank como ***corpus*** e valores de $n = 10$ e $n = 15$ para treinamento e teste, respectivamente. Para árvores sintáticas *gold*, o melhor sistema conjunto obteve 95,0% de F_1 para identificação, 91,4% de acurácia para classificação e 91,2% de F_1 para a tarefa combinada. Já para árvores sintáticas automáticas obtidas com o *parser* de Charniak e Johnson (2005), o melhor sistema conjunto obteve 83,4% de F_1 para identificação, 92,0% de acurácia para classificação e 80,0% de F_1 para a tarefa combinada.

e) Rosa e Adán-Coello (2010) propõem um sistema simbólico-conexionista[6] que, diferentemente dos sistemas anteriores, só emprega atributos semânticos e não sintáticos/lexicais. Como ***corpus*** de treinamento, empregam-se só sentenças bem formadas (sujeito-verbo-objeto) geradas automaticamente, acompanhadas pela estrutura argumental de cada verbo alvo da sentença (i.e., o algoritmo é supervisionado). O **conjunto de papéis semânticos** está composto de: `AGENTE`, `EXPERIENCIADOR`, `CAUSA`, `PACIENTE`, `TEMA`, `LOCAL` e `VALOR`. A representação de **atributos** das palavras usa a classificação de verbos (*body*, *change*, *communication*, etc.) e substantivos (*action*, *life*, *element*, etc.) da WordNet; e a representação clássica de micro-atributos semânticos de Waltz e Pollack (1985) e McClelland e Kawamoto (1986) (por exemplo, humano–não humano, suave–duro, masculino–feminino, etc.). O sistema atinge uma precisão de 94%, na tarefa combinada, para um conjunto de 120 verbos de sentenças de teste geradas automaticamente.

Os sistemas de APS até aqui apresentados foram desenvolvidos para o inglês. Mas também existem alguns trabalhos para outras línguas:

[6]Uma teoria temática simbólica é usada para fornecer à rede conexionista do conhecimento inicial.

1. **Sueco:** Johansson e Nugues (2006) usaram um *corpus* paralelo inglês-sueco, cuja seção inglesa foi anotada (automaticamente) com papéis semânticos usando o estilo da FrameNet, para derivar uma parte anotada em sueco. Usando esta anotação transferida como *corpus* de treinamento, implementaram um sistema de APS de duas fases (identificação e classificação) com SVMs como algoritmo de aprendizado. Usaram atributos convencionais como lema do predicado, POS do predicado, voz, posição, núcleo, etc. Obtiveram 75% de acurácia na tarefa de classificação de argumentos; e 67% e 47% de precisão e cobertura na tarefa combinada.

2. **Holandês:** Monachesi et al. (2007) desenvolveram um anotador baseado em regras para anotar um *treebank* em holandês com papéis semânticos, empregando o modelo do PropBank. Depois de corrigi-lo manualmente, este *corpus* foi usado no treinamento de um sistema de APS para holandês usando um algoritmo de aprendizado baseado em memoria (*Memory Based Learning*). Usaram-se atributos convencionais como voz, POS do constituinte, núcleo, POS do núcleo, etc. Este sistema obteve 70,3% de precisão, 70,6% de cobertura, e 70,4 de F_1 na tarefa combinada.

3. **Chinês:** Xue (2008) emprega o *corpus* PropBank Chinês para implementar um sistema de APS baseado em modelos de máxima entropia. Utiliza-se um sistema de 3 fases: poda, identificação e classificação de argumentos; assim como atributos convencionais: posição, subcategorização, tipo de sintagma, etc., mas alguns deles foram usados só na fase de classificação e outros em ambas. Quando são usadas árvores sintáticas *gold*, o sistema obtém 94,1% de acurácia para classificação de argumentos; e 93,0% de precisão, 91,0% de cobertura e 92,0 de F_1. Os autores também reportam resultados no NomeBank Chinês para predicados nominais.

4. **Árabe:** Diab et al. (2008) apresentam um sistema de APS para árabe moderno que explora os atributos morfológicos da língua. Está baseado em um modelo supervisionado que usa SVMs para identificação e classificação de argumentos. O sistema é treinado e testado no PropBank para Árabe, usando árvores sintáticas *gold*. Usaram-se atributos convencionais como lema do predicado, caminho, caminho parcial, tipo de sintagma, etc.; e outros específicos do árabe como morfologia flexional (número, gênero, caso, etc.) e derivacional (lema das palavras com todos os diacríticos indicados). Na tarefa combinada, o sistema atinge 82,2 de F_1.

5. **Espanhol e Catalão:** Morante e Bosch (2009) experimentam com diferentes transformações de atributos em um sistema de APS para espanhol e catalão. Experimentam com os dados fornecidos por Màrquez et al. (2007), em conjuntos de dados dentro e fora de um mesmo domínio. Utilizam um classificador baseado em memoria, em um sistema de duas fases: identificação e classificação de argumentos.

Empregam um conjunto de 88 atributos usados em outros sistemas de APS, como: forma e lema do verbo, tipo de sintagma, identidade da preposição, etc. Além disso, criaram-se novos atributos dividindo e combinando alguns dos atributos anteriores: lema do verbo + preposição do constituinte, lema do verbo + preposição + função sintática do constituinte, etc. Os resultados obtidos na tarefa combinada são: 88,9 e 85,3 de F_1 para catalão no mesmo e diferente domínio, respectivamente; e 84,0 e 87,4 de F_1 para espanhol no mesmo e diferente domínio, respectivamente.

Para uma revisão mais extensa dos sistemas baseados em *corpus* desenvolvidos para APS automática, recomenda-se Moreda Pozo (2008) e Màrquez et al. (2008).

3.3.2 Sistemas Não Supervisionados

a) Abend et al. (2009) focam na sub-tarefa de identificação de argumentos, usando um algoritmo que precisa somente da anotação de *part-of-speech*, assim como um *parser* sintático totalmente não supervisionado. O sistema foi testado no *corpus* PropBank para o inglês e o espanhol. O melhor modelo obtém uma precisão de 55,97% e F_1 de 59,14 para o inglês; e precisão de 21,8% e F_1 de 23,87 para o espanhol.

b) Abend e Rappoport (2010) trabalham na classificação de argumentos principais e adjuntos. Para isso, utilizam indução não supervisionada de gramáticas e algoritmos de indução de *part-of-speech*, com foco em argumentos preposicionados. Avaliam o método com o *corpus* PropBank, obtendo em torno de 70% de acurácia quando avaliados com argumentos preposicionados e mais de 80% para todo o conjunto de argumentos.

c) Lang e Lapata (2010) utilizam um método para classificação de argumentos (por eles chamado de indução de papéis), baseado na ideia de detectar alternâncias sintáticas e encontrar sua forma canônica. Isto é implementado usando um modelo probabilístico que é uma variação de um classificador logístico. O modelo é treinado só com informação sintática obtida usando um *parser* automático. Usando o *corpus* PropBank do CoNLL-2008[7] (Surdeanu et al., 2008a), o sistema obteve uma *purity*[8] de *clusters* de 82,6% e F_1 de 76,1%, o que representa um ganho de 8,7% em *purity* e 13% em F_1 sobre um *baseline* que agrupa as instâncias só baseado nas suas etiquetas sintáticas.

d) Lang e Lapata (2011a) apresentam um algoritmo para indução de papéis que, desde uma partição inicial dos dados, mescla iterativamente *clusters* que representam papéis semânticos, assim levando um agrupamento inicial a um final de melhor qualidade. O agrupamento inicial é executado com base em uma medida de similaridade sintática.

[7] Explicado na Seção 3.3.4.
[8] Porcentagem de instâncias que pertencem à classe *gold* majoritária no *cluster*.

O processo iterativo restante baseia-se em 3 medidas de similaridade: lexical, *part-of-speech*, e *frame*. Complementa-se o algoritmo com um conjunto de regras para realizar identificação de argumentos. Novamente usando o *corpus* PropBank do CoNLL-2008, e testando com combinações árvores sintáticas *gold*/automáticas e indentificação de argumentos *gold*/automáticas, os resultados são: *purity* de 81,9% e F_1 de 76,2 para auto/auto; *purity* de 84,0% e F_1 de 78,9 para gold/auto; *purity* de 86,5% e F_1 de 77,3 para auto/gold; e *purity* de 88,7% e F_1 de 80,1 para gold/gold.

e) Lang e Lapata (2011b) implementam um método para indução de papéis baseado em particionamento de grafos. Dado um verbo, o algoritmo constrói um grafo com pesos cujos vértices correspondem aos argumentos do verbo, e as arestas com pesos quantificam a similaridade entre as instâncias. O grafo é particionado em *clusters* que representam os papéis semânticos. O algoritmo iterativamente atribui etiquetas de *clusters* aos vértices do grafo, através da seleção da etiqueta mais comum entre os seus vizinhos. Novamente usando o *corpus* PropBank do CoNLL-2008, e testando com combinações árvores sintáticas *gold*/automáticas e identificação de argumentos *gold*/automáticas, os resultados são: *purity* de 82,5% e F_1 de 75,0 para auto/auto; *purity* de 84,0% e F_1 de 78,4 para gold/auto; *purity* de 87,4% e F_1 de 75,2 para auto/gold; e *purity* de 88,6% e F_1 de 78,6 para gold/gold.

Aperfeiçoamentos destes dois últimos métodos, como apresentados em Lang (2012), são estudados em maior detalhe no Capítulo 5.

3.3.3 Sistemas Semissupervisionados

a) He e Gildea (2007) investigam dois **algoritmos** semissupervisionados – *co-training* e *self-training* – que, começando com um conjunto pequeno de dados anotados e um ou dois classificadores "fracos", visam melhorar o desempenho do sistema incorporando dados não anotados no conjunto de treinamento. Adota-se a definição de *self-training* segundo Clark et al. (2003): é um procedimento no qual "um anotador é re-treinado na sua própria *cache* anotada em cada iteração". *Co-training* (Blum e Mitchell, 1998) emprega dois classificadores treinados em duas "vistas" dos dados (i.e., subconjuntos de atributos disjuntos) que podem se ajudar entre si, adicionando seus exemplos mais confiáveis no conjunto de treinamento de cada um. Nos experimentos, utilizaram modelos de Máxima Entropia (para *self-training*) e Listas de Decisão (para ambos). Só usaram os **atributos Núcleo** e **Caminho**. Em geral, os resultados obtidos foram muito pobres. Usando a FrameNet, o seu melhor sistema obteve um F_1 em torno de 33 para a tarefa combinada.

b) Fürstenau e Lapata (2012) visam melhorar o desempenho de um sistema supervisionado ampliando seu conjunto de dados de treinamento com anotações automaticamente inferidas de dados não anotados. A ideia central é descobrir instâncias novas para treinar o classificador, com base na sua similaridade com as instâncias anotadas iniciais. A motivação é que as sentenças que são similares no seu léxico e na sua estrutura sintática têm alta probabilidade de compartilhar uma análise semântica de *frames*. As sentenças são representadas como grafos de dependências e procura-se um alinhamento (estrutural) ótimo entre eles, para depois projetar as anotações semânticas. Os grafos são pontuados usando uma função baseada em similaridade lexical e sintática. Obtém-se o alinhamento de grafos com melhor pontuação usando programação linear. Utilizando diferentes subconjuntos do ***corpus*** da FrameNet como dados de treinamento iniciais, e sentenças não anotadas do *British National corpus*, realizam-se experimentos usando de 1 a 6 sentenças de expansão inferidas automaticamente para um *corpus* de treinamento inicial de 1 a 10 sentenças por verbo anotadas manualmente. O desempenho do sistema (na tarefa combinada) melhora para valores intermediários do número de sentenças de expansão, com acréscimos em F_1 desde 11,61% até 12,82%.

c) Zadeh Kaljahi (2010) enfrenta o problema de propagação de ruído na classificação de argumentos, usando métodos de balanceamento e pré-seleção para *self-training* (Yarowsky, 1995) com modelos de Máxima Entropia. Emprega-se uma **estratégia** de duas etapas: poda dos candidatos a argumentos que são menos prováveis; e identificação e classificação conjunta de papéis semânticos. Os **atributos** utilizados são :

- **Tipo de Sintagma**, **Lema do Verbo**, **POS do Verbo**, **Caminho**, **Lema do Núcleo**, **POS do Núcleo**, **Categoria Principal** e **Subcategorização:** como descritos previamente.
- **Posição + Voz:** Concatenação dos atributos **Posição** e **Voz** como descritos antes.
- **POS do Conteúdo:** *part-of-speech* do **Núcleo do Sintagma Preposicional**.
- **Subcategorização do Constituinte:** igual que **Subcategorização**, mas para o constituinte a ser anotado.
- **Contas no Caminho:** número de orações, sintagmas nominais e sintagmas verbais no **Caminho**.
- **Distância:** número de palavras entre o constituinte e o verbo alvo.
- **Identificador de Verbo Composto:** indica se o verbo alvo é `simples`, `composto`, ou `composto descontínuo`.

- **Posição do Núcleo no Constituinte:** número de palavras à direita e esquerda do núcleo dentro do constituinte.

Propõem-se duas modificações ao algoritmo de *self-training*. Por um lado, como o classificador base é relativamente "fraco" pelo tamanho reduzido do *corpus* inicial de treinamento, **pré-selecionar**, em cada iteração, um conjunto de exemplos não anotados que seja mais provável de estar corretamente etiquetado pelo classificador nos passos iniciais. Para isso, usa-se uma medida de simplicidade de sentenças baseada no número de candidatos a argumentos extraídos de cada sentença: maior o número de candidatos extraídos, menor a simplicidade da sentença. Por outro lado, **balancear** o novo conjunto de dados anotados a ser adicionados ao dados de treinamento. Propõe-se uma forma de balanceamento baseada na distribuição dos papéis na sentença. Como medida para selecionar uma sentença anotada, usa-se a média das probabilidades atribuídas pelo classificador a todos os argumentos extraídos da sentença.

Quando treinado no conjunto de dados não anotados do WSJ, o método balanceado obteve um desempenho (comparado em F_1) muito melhor do que o não balanceado, tanto no WSJ (68,5 vs. 67,9) quanto nos conjuntos de teste do *corpus* Brown (59,6 vs. 58,9). Além disso, entre as duas estratégias de pré-seleção, o método baseado em simplicidade obtém um desempenho melhor do que o aleatório (59,7 vs. 59,3).

3.3.4 Competições Internacionais

Considerando o crescente interesse no estudo dos papéis semânticos e dos sistemas de anotação automática dos mesmos, foram propostas várias conferências com o único objetivo de criar um foro específico, no qual se possa discutir e comparar resultados e experiências. Entre estas conferências destacam-se a CoNLL[9] (Carreras e Màrquez, 2004, 2005; Surdeanu et al., 2008a; Hajic et al., 2009); e o Senseval/SemEval (Litkowski, 2004; Màrquez et al., 2007) com as suas respectivas *Shared Tasks*. A seguir, estas competições são descritas em ordem cronológica.

Senseval-3 (2004) Task: Anotação Automática de Papéis Semânticos

A tarefa consistia em realizar APS automática para o **inglês** usando os dados da **FrameNet**. O desafio básico foi: dada uma sentença, o predicado alvo e o seu *frame*, identificar os *frame elements* dentro da sentença e anotá-los com os nomes apropriados de *frame elements*.

Usaram-se 8.002 sentenças selecionadas aleatoriamente de 40 *frames* (também selecionados aleatoriamente) que tinham pelo menos 370 anotações (dos 100 *frames* que tinham

[9]Conference on Computational Natural Language Learning

a maior quantidade de anotações). Os sistemas participantes podiam utilizar qualquer e toda a informação nos dados da FrameNet para treinamento e desenvolvimento.

Os sistemas foram avaliados usando as medidas de precisão e cobertura de *frame elements* e sobreposição das posições na sentença dos *frame elements* anotados pelo sistema e aqueles identificados nos dados da FrameNet. Participaram 8 equipes, obtendo uma precisão média de 80,3% (que é um pouco menor a 82% atingido por Gildea e Jurafsky (2002)), e cobertura média de 75,7%. Muitas equipes atingiram uma precisão igual ou maior a 90% que indica que as suas implementações para classificação de constituintes são bastante boas.

CoNLL 2004 - Shared Task: Anotação de Papéis Semânticos

A tarefa consistia em APS **por constituintes para o inglês**, considerando predicados verbais. O *corpus* usado foi o **PropBank** – a versão liberada em fevereiro de 2004. O desafio foi criar estratégias de AM para o problema de APS na base de **informação sintática parcial**, evitando o uso de árvores sintáticas completas e bases de conhecimento léxico-semânticas externas. A informação fornecida para cada sentença inclui: palavras, etiquetas *part-of-speech*, *chunks* em formato IOB2[10], orações em formato Início-Fim, entidades nomeadas em formato IOB2, verbos alvo (forma base), e os papéis semânticos dos argumentos do verbo alvo em formato Início-Fim. Os sistemas foram avaliados com respeito à precisão, cobertura e F_1. Para que um argumento seja reconhecido como correto, as palavras que formam parte dele assim com o papel semântico atribuído devem ser corretos. Algumas lições aprendidas a considerar são:

- A maioria dos sistemas tratou a anotação dos argumentos de cada verbo em uma sentença como um problema independente.
- A estratégia predominante foi de duas fases: reconhecimento/poda/identificação e classificação de argumentos. Isto implica trabalhar com candidatos a argumentos na segunda fase, o que permite desenvolver atributos para argumentos completos.
- Todos os sistemas participantes realizaram o aprendizado com classificadores independentes do verbo. A informação que poderia ser fornecida pelo verbo alvo é capturada através de atributos e algumas restrições globais.
- Sobre a granularidade na qual os elementos da sentença são processados, tornou-se muito claro que uma boa eleição para este problema é o processamento sintagma

[10]Palavras fora de um *chunk* recebem a etiqueta `O`. Para as palavras que formam um *chunk* de tipo k, a primeira recebe a etiqueta `B-k` (*Begin*), e as restantes recebem a etiqueta `I-k` (*Inside*)

por sintagma, porque os limites de um sintagma normalmente coincidem com os limites dos argumentos.

- Alguns sistemas usaram algum tipo de pós-processamento para garantir coerência na anotação final, corrigir alguns erros do sistema, ou tratar alguns tipos de argumentos adjuntos. Na maioria dos casos, este processo é realizado com um conjunto de regras simples.

Dos 10 sistemas participantes, Hacioglu et al. (2004) obtiveram os melhores resultados, com um desempenho moderado de 69,49 em F_1. O sistema utiliza SVMs como algoritmo de aprendizado, tomando decisões IOB nos *chunks* das sentenças, e explorando uma ampla variedade de atributos baseado em análise sintática parcial.

CoNLL 2005 - Shared Task: Anotação de Papéis Semânticos

Como no ano 2004, esta edição tratou sobre o reconhecimento de papéis semânticos **por constituintes para o inglês**, mas com algumas novidades introduzidas:

- Árvores sintáticas completas geradas pelos *parsers* de Collins (1999) e de Charniak (2000), para avaliar a contribuição de informação sintática completa.

- Um *corpus* maior de treinamento (PropBank), para testar a escalabilidade dos sistemas de APS baseados em AM.

- Dados de teste do *corpus* Brown anotados seguindo o modelo PropBank, para testar a robustez dos sistemas apresentados com uma avaliação *cross-corpora*.

Foram concebidos dois tipos de avaliações: desafio fechado, se os sistemas usam só a informação dos dados de treinamento; e desafio aberto, se é usado algum tipo de informação ou recurso externo. Dezenove sistemas participaram do desafio fechado, e nenhum no aberto. Algumas lições aprendidas a considerar são:

- Aproximadamente 8 diferentes algoritmos de aprendizado foram aplicados no treinamento dos sistemas. Modelos logarítmico-lineares e classificadores lineares baseados em vetores dominaram os demais. Em particular, 8 equipes usaram modelos de Máxima Entropia e 6 empregaram SVMs.

- Muitos sistemas usaram algum tipo de combinação de sistemas para incrementar sua robustez e cobertura. As saídas para combinar são obtidas mudando a informação de entrada, trocando o algoritmo de aprendizado, ou considerando uma lista de n melhores soluções.

- A maioria dos sistemas empregou anotação sobre os nós das árvores sintáticas, procurando um mapeamento um-a-um entre os argumentos e os constituintes da árvore.
- A maioria dos sistemas empregam uma estratégia de 4 fases: poda, identificação, classificação, e pós-processamento.
- Os principais tipos de atributos usados nesta edição podem ser divididos em quatro categorias gerais:

    1. Atributos que caracterizam a estrutura do argumento candidato: pai e irmãos do argumento (tipo sintático e núcleo), *tokens* à direita e esquerda do argumento, etc.
    2. Atributos que descrevem propriedades do predicado realizado por verbo que é foco de análise: forma, lema, etiqueta de função gramatical, voz, subcategorização, etc.
    3. Atributos que capturam a relação entre o predicado realizado por verbo e o constituinte que vai ser anotado: posição relativa entre eles, distância entre eles (baseada no número de palavras), caminho na árvore, etc.
    4. Atributos globais que descrevem a anotação completa dos argumentos do predicado: o padrão sequencial dos argumentos do predicado.

- Todos os sistemas experimentaram uma severa queda em desempenho (quase 10 pontos em F_1) no conjunto de dados de teste Brown.

Houve sete sistemas com um desempenho F_1 final entre 75 e 78, mais sete com desempenhos entre 70 e 75, e 5 com um desempenho entre 65 e 70. O melhor sistema foi Punyakanok et al. (2005) que atingiu quase 79,4 em F_1 no conjunto de teste do WSJ, 67,8 nos dados de teste do Brown, e 77,9 no teste combinado (WSJ + Brown).

Os melhores resultados nesta edição foram 10 pontos melhores do que aqueles da versão prévia. Este acréscimo no desempenho pode ser atribuído à combinação do seguintes fatores: (i) os conjuntos de treinamento foram significativamente maiores; (ii) árvores sintáticas completas foram disponibilizadas como informação de entrada; e (iii) esquemas mais sofisticados de combinação foram implementados.

SemEval-2007 Task 9: Anotação Semântica Multi-nível de Catalão e Espanhol

Visou avaliar e comparar sistemas automáticos para anotação semântica em diferentes níveis para o **catalão e o espanhol**. Os três níveis semânticos considerados incluem:

papéis semânticos e desambiguação verbal, desambiguação de todos os substantivos, e reconhecimento de entidades nomeadas. A anotação de papéis semânticos de predicados verbais segue o estilo do PropBank, e a tarefa é similar à indicada no CoNLL 2005 *Shared Task*. Desambiguação verbal refere-se à atribuição da etiqueta de *role set* apropriada.

O *corpus* usado é um subconjunto do CESS-ECE, um *Treebank* multi-língua, composto de um *corpus* de espanhol (CESS-ESP) e catalão (CESS-CAT) de 500.000 palavras cada um (Martí e Taulé, 2007). Este *corpus* foi enriquecido com diferentes tipos de informação semântica: estrutura de argumentos, papéis semânticos, classe semântica, entidades nomeadas, e *synsets* da WordNet para os 150 substantivos mais frequentes. O processo de anotação foi semi-automático, com uma revisão manual após todos os processos automáticos. O *corpus* foi dividido em treinamento e teste com uma proporção 90%-10%, assim com em dois subconjuntos em-domínio e fora-de-domínio.

O formato dos dados é igual ao do CoNLL 2004/2005 e fornecem a seguinte informação: palavra, substantivo alvo, verbo alvo, lema, *part-of-speech*, análise sintática completa, entidades nomeadas, sentido na WordNet do substantivo alvo, classe semântica do verbo, e argumentos com papéis semânticos.

Dos dois únicos sistemas participantes, ILK2 (Morante e Busser, 2007) obteve os melhores resultados para APS: 83,4 de F_1 para catalão e 84.1 para espanhol. Este sistema emprega classificação baseada em memória de constituintes sintáticos, usando um conjunto variado de atributos.

CoNLL 2008 - Shared Task: Anotação Conjunta de Dependências Sintáticas e Semânticas

Propõe um formalismo unificado baseado em dependências, que modela dependências sintáticas e semânticas. Conceitualmente, esta *task* pode ser dividida em três: (i) análise sintática de dependências, (ii) identificação e desambiguação de predicados semânticos, e (iii) identificação de argumentos e atribuição de papéis semânticos para cada predicado. Pela complexidade que apresenta, esta *task* só foi realizada para o inglês. Como no CoNLL 2005, a avaliação foi dividida em desafio aberto e fechado.

Os dados de entrada possuem a seguinte informação: número de *token*, palavra, lema, *gold part-of-speech*, *part-of-speech* automática, *tokens* divididos no hífen, lema automático do *token* dividido, *part-of-speech* automático do *token* dividido, núcleo, relação de dependência sintática, *role sets* dos predicados da sentença, e argumentos dos predicados.

O *corpus* usado para treinamento e teste foi gerado através de um processo que combina vários *corpora* (Penn TreeBank, PropBank, NomBank) e os converte de um formato baseado em constituintes a dependências (ver Surdeanu et al. (2008a) para obter detalhes sobre este processo).

Dos 19 sistemas participantes, os melhores resultados foram obtidos por Johansson e Nugues (2008): F_1 de 80,37 no conjunto de dados de teste WSJ+Brown, 81,75 no WSJ, e 69,06 no Brown para o desafio fechado. Este sistema emprega métodos estado-da-arte para cada uma das subtarefas: modelo de análise sintática de segunda ordem; modelos de identificação e classificação de argumentos separados, especialmente implementados para PropBank e NomBank; inferência com *re-ranking* para APS; e, finalmente, otimização conjunta de todas as tarefas usando *meta-learning*[11].

CoNLL 2009 - Shared Task: Dependências Sintáticas e Semânticas em Múltiplas Línguas

Esta edição teve o mesmo objetivo que no ano 2008, mas agora para mais 6 línguas (catalão, chinês, tcheco, alemão, japonês e espanhol) além do inglês. Os participantes deviam escolher entre duas tarefas:

- **Tarefa Conjunta:** análise sintática de dependências e APS.
- **Só APS:** forneciam-se árvores sintáticas de dependências usando *parsers* estado-da-arte para cada língua.

Os dados de teste indicavam para quais predicados devia ser feita a anotação para a tarefa de APS. Os **desafios fechado e aberto** da edição anterior foram mantidos; os participantes podiam escolher um ou os dois desafios. No desafio fechado, os sistemas deviam ser treinados estritamente com a informação contida no *corpus* de treinamento fornecido; no desafio aberto, os sistemas podiam ter sido desenvolvidos usando qualquer tipo de ferramenta ou recurso externo.

Os dados de entrada possuem a seguinte informação: número de *token*, palavra, lema, lema automático, *gold part-of-speech*, *part-of-speech* automática, atributos morfológicos *gold*, atributos morfológicos automáticos, núcleo *gold*, núcleo automático, relação de dependência sintática *gold*, relação de dependência sintática automática, *role sets* dos predicados da sentença, e argumentos dos predicados. Dependendo da língua, alguns destes dados podem não estar disponíveis.

Algumas estatísticas e resultados importantes são apresentadas a seguir[12]:

- **Tarefa Conjunta:** Participaram 13 sistemas; 11 no desafio fechado e 2 no aberto. Tanto para o desafio fechado quanto para o aberto, os melhores resultados foram de

[11] A principal diferença com o aprendizado "base" está no âmbito do nível de adaptação. Enquanto o aprendizado no nível "base" tem como foco acumular experiência em uma tarefa de aprendizado específica, o aprendizado no "meta-nível" se preocupa com acumular experiência sobre o desempenho de múltiplas aplicações de um sistema de aprendizado

[12] Resultados mais detalhados (como tabelas de precisão e cobertura para a tarefa de APS, etc.) estão disponíveis em `http://ufal.mff.cuni.cz/conll2009-st/results/results.php`.

Che et al. (2009), que obtiveram uma média para todas as línguas de 82.64 de F_1 para o primeiro caso e de 82.70 de F_1 para o segundo. Para a análise sintática de dependências, utiliza-se um modelo pseudo-projetivo baseado em grafos de ordem superior; para classificar os sentidos do predicado alvo, usa-se um modelo SVM; e para APS emprega-se um modelo de Máxima Entropia junto com programação linear inteira.

- **Só APS:** Participaram 7 sistemas, todos no desafio fechado. O melhor sistema foi de Zhao et al. (2009), que obtiveram uma média de 80.47 de F_1 para todas as línguas. O sistema utiliza modelos de Máxima Entropia para todas as subtarefas de classificação. Diferente da maioria de sistemas, as etapas de identificação e classificação de argumentos são realizadas de forma conjunta.

3.4 Anotação Automática de Papéis Semânticos e Tarefas Relacionadas para o Português

Na seção anterior, foram discutidos trabalhos realizados para APS automática, a maioria na língua inglesa. Contudo, para a língua portuguesa não existem muitos trabalhos que tenham explorado o processo de anotação semântica automática de textos, principalmente por não existirem os recursos lexicais necessários.

a) Rosa (2007) apresenta um sistema híbrido simbólico-conexionista, antecessor de Rosa e Adán-Coello (2010) para o português. O sistema possui dois módulos: um *parser* simbólico baseado em eventos que emprega uma gramática que toma em consideração classes de advérbios, verbos transitivos e não transitivos; e um preditor biologicamente plausível conexionista de estruturas predicado-argumento. Usam-se os mesmos micro-atributos semânticos, etiquetas de papéis semânticos e forma de treinamento e teste que em Rosa e Adán-Coello (2010). O sistema atinge 94% de precisão e cobertura na tarefa combinada, para um conjunto de 120 verbos de sentenças de teste geradas automaticamente.

b) Bick (2007) descreve um método para APS de sentenças em português empregando uma gramática com 500 regras de restrição escritas manualmente, além de explorar as relações de dependência sintática, assim como os protótipos de classes semânticas e funções sintáticas. Foram desenvolvidos experimentos em textos em português europeu (seção CETENPúblico do Bosque), atingindo uma cobertura de 86.6% e uma precisão de 90.5%.

c) Sequeira et al. (2012) implementam um sistema de APS baseado em *corpus* para português europeu. Anotou-se automaticamente a Seção CETEMPúblico do *corpus* Bosque com etiquetas `P` (predicado), `ARG0` (agente prototípico) e `ARG1` (paciente prototípico) de acordo com as categorias sintática dos constituintes (verbo, sujeito e objeto, respectivamente). Estas sentenças anotadas automaticamente, foram usadas como dados de treinamento para dois classificadores (SVMs e CRF). O melhor classificador (SVM) obtém 31.1 de F_1 na anotação de `ARG0` e 19.0 para `ARG1`. Aparentemente, os dados de treinamento não foram validados manualmente, o que poderia ser a causa dos baixos resultados obtidos.

d) Para o português do Brasil, existem duas propostas para desenvolver sistemas de APS usando o *corpus* PropBank.Br. Alva-Manchego e Rosa (2012b) propõem uma abordagem semissupervisionada usando o algoritmo *self-training* com modelos de Máxima Entropia. Fonseca e Rosa (2012) descrevem um arquitetura de redes neurais capaz de executar diferentes tarefas de PLN, entre elas APS. Até o momento da escrita deste documento, não existem resultados publicados relacionados com estas propostas.

e) Amancio et al. (2010) apresentam um sistema para anotação automática com etiquetas de perguntas – `quem`, `como`, `com o quê`, etc. – aos argumentos de verbos em sentenças simplificadas para o português. O *corpus* usado contém 104 artigos de notícias do jornal brasileiro Zero Hora, que foram simplificadas manualmente no projeto PorSimples (Caseli et al., 2009). Este *corpus* foi anotado manualmente com etiquetas de perguntas, para depois ser usado como conjunto de treinamento para um classificador para esta tarefa. Usam-se atributos convencionais como: tipo de sintagma, posição, verbo simples ou composto, etc. Nos testes realizados, o melhor sistema obtém 79 de F_1 usando o algoritmo SMO (*Sequential Minimal Optimization* – um algoritmo para treinar SVMs).

3.5 Considerações Finais

Neste capítulo foram analisadas as diferentes abordagens empregadas para automatizar a tarefa de anotação de papéis semânticos. A grande maioria delas emprega técnicas de aprendizado de máquina (supervisionado) para treinar um classificador que será o encarregado da tarefa de anotação. Estas abordagens foram descritas com base nos algoritmos que elas empregam, e com especial ênfase nos atributos dos constituintes das sentenças que permitem determinar as etiquetas semânticas que cada um deles possui.

Os *Shared Tasks* da CoNLL e Senseval/SemEval foram grandes impulsores de pesquisa em APS, porque forneciam um esquema único padrão (*benchmark*) para avaliar sistemas

para esta tarefa de PLN. Muitos sistemas desenvolvidos posteriormente empregam os recursos e métricas destas competições para avaliar o seu desempenho e compará-lo com os de outros sistemas.

Finalmente, foram descritos as pesquisas sobre anotação de papéis semânticos (e tarefas relacionadas) existentes para a língua portuguesa e foi evidenciado que, em comparação ao realizado para textos em língua inglesa, a pesquisa em APS para o português é reduzida. Uma clara consequência deste fato é que não existe sistema que sirva como comparação de desempenho do implementado neste projeto. No capítulo seguinte indica-se como este problema foi resolvido.

Capítulo

4

Benchmark de Comparação e um Sistema Supervisionado

Para avaliar o desempenho e qualidade de um sistema, é comum compará-lo com outros em igualdade de condições. No caso da APS automática, isto envolve usar um mesmo conjunto de dados de treinamento e teste, assim como igual metodologia de avaliação. Para o português do Brasil, não existem sistemas de APS automática com os quais o desenvolvido neste trabalho possa ser comparado: Bick (2007) trabalhou com o português de Portugal, e o *corpus* usado não está disponível; Sequeira et al. (2012) também desenvolveu para o português de Portugal, e o *corpus* utilizado não é confiável porque não foi revisto manualmente; finalmente, Fonseca e Rosa (2012), embora foquem no português do Brasil, ainda não disponibilizaram resultados da sua pesquisa.

Neste cenário, decidiu-se implementar um *benchmark* próprio para comparação e avaliação, baseado nas CoNLL *Shared Tasks* (STs) de APS automática baseada em constituintes (Carreras e Màrquez, 2004, 2005), amplamente usadas para comparar sistemas de APS para o inglês. Os recursos fornecidos são conjuntos de dados de treinamento e teste (Seção 4.1), métricas apropriadas de avaliação (Seção 4.2), e um sistema *baseline* baseado em regras simples, útil para uma comparação básica (Seção 4.3).

Além disso, implementou-se um sistema supervisionado usando os recursos disponibilizados no *benchmark* (Seção 4.4). Este sistema estende o trabalho de Alva-Manchego e Rosa (2012a) usando um conjunto maior de atributos dos constituintes das sentenças e experimentando com um algoritmo de aprendizado mais sofisticado como Regressão Logística (também conhecido como Máxima Entropia).

4.1 Conjuntos de Dados

Sentenças com informação de estrutura predicado-argumento foram extraídas do *corpus* PropBank.Br. Como nas STs, usa-se uma representação plana em colunas para as anotações de cada sentença. Cada coluna contém algum tipo de anotação, associando uma etiqueta com cada palavra. A Tabela 4.1 explica a informação fornecida para cada sentença, e a Fig. 4.1 apresenta um exemplo de uma sentença completamente anotada[1].

Tabela 4.1: Informação de cada coluna. Os campos acima de 9 não estão disponíveis no conjunto de teste.

Número	Nome	Descrição
1	`ID`	Contador de *tokens* que inicia em 1 para cada nova sentença
2	`FORM`	Forma da palavra ou sinal de pontuação
3	`LEMMA`	Lema *gold-standard* da `FORM`
4	`GPOS`	Etiqueta *part-of-speech gold-standard*
5	`FEAT`	Atributos morfológicos *gold-standard*
6	`CLAUSE`	Orações em formato início-fim
7	`FCLAUSE`	Orações com informação de tipo em formato início-fim
8	`SYNT`	Árvore sintática *gold-standard* completa
9	`PRED`	Predicados semânticos na sentença
10...	`ARG`	Colunas com etiquetas de argumentos para cada predicado semântico seguindo a ordem textual

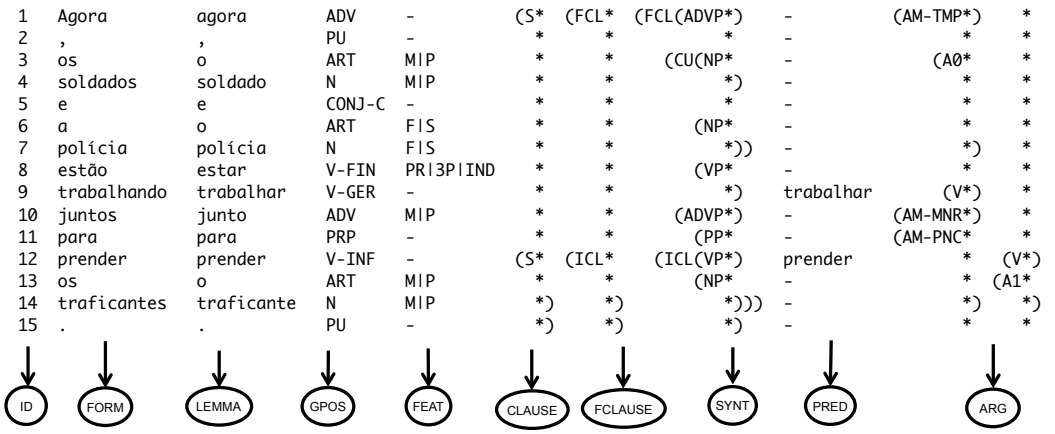


Figura 4.1: Sentença anotada do PropBank.Br no formato plano de colunas.

[1]Embora o *corpus* Bosque foi manualmente corrigido, ele ainda pode apresentar erros. Por exemplo, na sentença apresentada a palavra `juntos` deveria ser adjetivo (`ADJ`) e não adverbio (`ADV`).

4.1.1 *Processo de Conversão*

As sentenças no *corpus* PropBank.Br estão no formato Tiger-XML. Quando foram convertidas na representação plana por colunas descrita previamente, 1.331 proposições foram descartadas pelos seguintes motivos:

1. **Wrongsubcorpus:** Durante a anotação manual, as proposições no *corpus* que possuiam algum tipo de erro que atrapalhe a sua anotação com papéis semânticos, receberam a etiqueta `WRONGSUBCORPUS`. São três os tipos de erros existentes: erro de *parser* ou inadequação (por exemplo, um `NP` interno não anotado), erro de *corpus* (erro de ortografia, erro de pontuação, sentença fragmentada) e erro de evocação de verbo (verbo auxiliar ou adjetivo na forma de particípio). Por exemplo, na árvore sintática da Fig. 4.2, existe um erro de *parser* porque ela não possui um constituinte que corresponda ao sujeito do verbo composto diz respeito. No total, 312 proposições (ou instâncias de anotação) foram descartadas do *corpus* por este motivo.

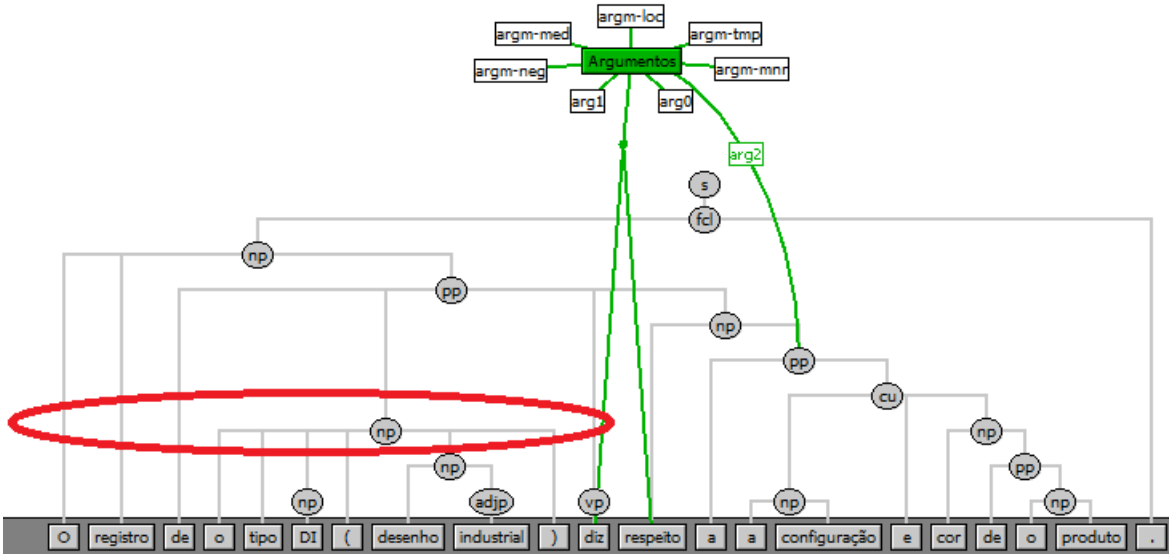


Figura 4.2: Exemplo de proposição com etiqueta `WRONGSUBCORPUS` (erro de *parser*) no *corpus* PropBank.Br.

2. **Árvore sintática com erros:** Elaborou-se um *script* para verificar se todos os nós da árvore sintática de cada proposição estão apropriadamente conectados (todos descendem do nó raiz). Esta verificação permitiu encontrar casos como o apresentado na Fig. 4.3, onde o nó `CU`, que corresponde a crônica e alta, não está ligado ao resto da árvore. Instâncias com este tipo de erro não foram anotadas como `WRONGSUBCORPUS` porque não atrapalhavam a anotação manual com papéis semânticos. Contudo, elas não podem ser consideradas no *corpus* de treinamento/teste, porque deseja-se

ter apenas árvores sintáticas corretas. No total, 16 instâncias foram descartadas por este motivo.

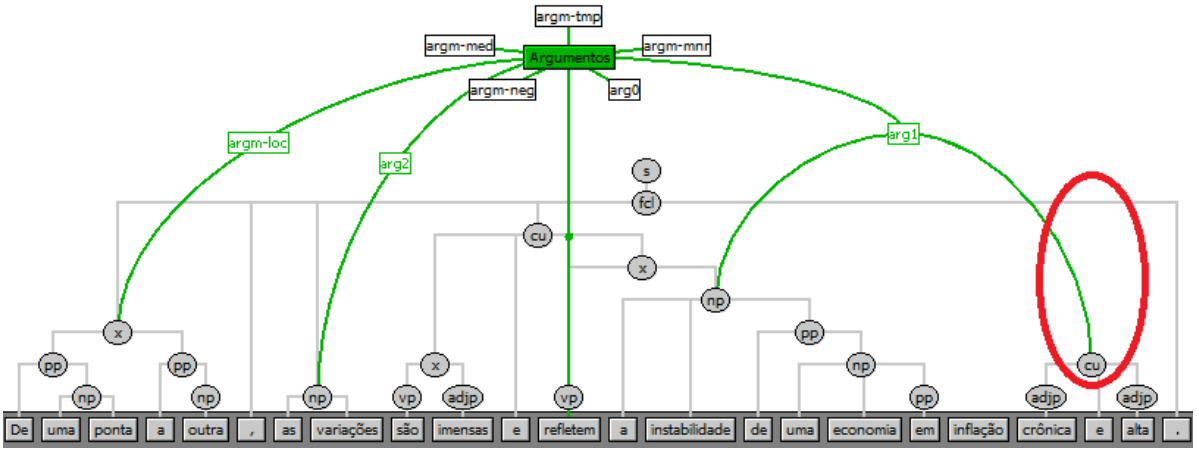


Figura 4.3: Exemplo de instância com árvore sintática com erros.

3. **Instâncias do verbo ser:** Seguindo a regras de anotação do projeto PropBank, as instâncias do verbo ser não possuem anotação com papéis semânticos no *corpus* PropBank.Br. No total, 964 instâncias foram descartadas por este motivo.

4. **Estrutura de Argumentos Incompleta:** Foram descartadas 25 instâncias nas quais não foram anotados os argumentos do verbo alvo, embora existiam os constituintes correspondentes.

5. **Argumentos Embutidos:** Por regra, não devem existir argumentos (com papel semântico) embutidos um no outro. Contudo, no *corpus* foram encontradas 14 instâncias nas quais isto acontece, pelos seguintes motivos:

   - **Erro na Anotação:** O papel semântico foi atribuído a um constituinte que não correspondia. No total, 7 instâncias foram descartadas por este motivo.
   - **Erro por Elipse:** As regras de anotação indicam que, nos casos de elipse, o argumento correferente ao constituinte omitido devia ser anotado. Como consequência, em 7 instâncias isto resultou em argumentos embutidos. Na Fig. 4.4, o verbo alvo `apresentar` atribui o papel `ARG0` a um constituinte cujo correferente é o `NP eu`. Assim, pela regra de anotação, este é anotado como `ARG0` e, consequentemente, fica embutido no `ARGM-ADV`.

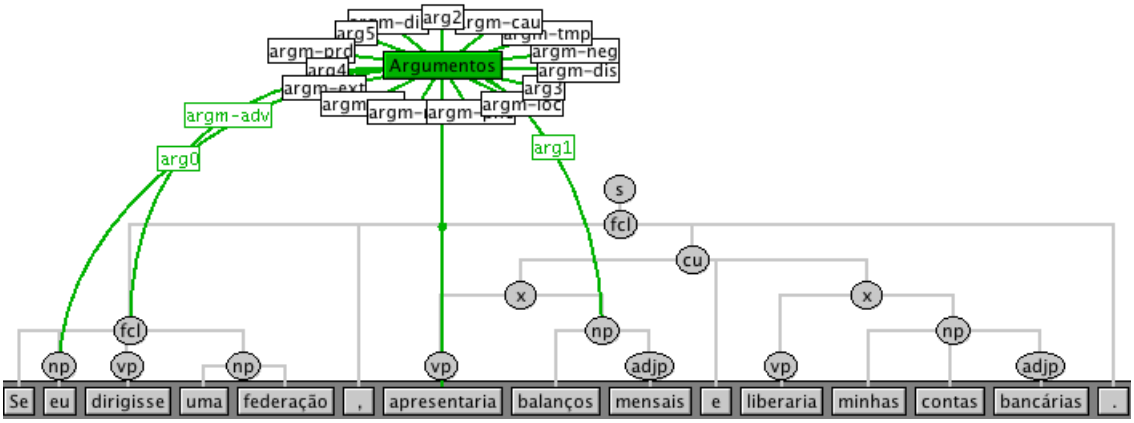


Figura 4.4: Exemplo de instância com argumentos embutidos por erro de elipse.

4.1.2 *Conjuntos de Treinamento e Teste*

As STs fornecem conjuntos de dados para treinamento, desenvolvimento e teste. Considerando a pouca quantidade de proposições anotadas disponíveis após o filtro descrito na seção anterior, decidiu-se dividir o conjunto total de dados só em treinamento e teste. Para realizar esta divisão, tomou-se como referência a CoNLL-X ST em análise sintática de dependências multi-língua (Buchholz e Marsi, 2006).

Nessa ST, a versão 7.3 do *corpus* Bosque foi apropriadamente dividida em treinamento e teste, cumprindo os requerimentos dos organizadores. Assim, usaram-se as mesmas sentenças para cada um dos nossos conjuntos de dados. As novas sentenças que aparecem no *corpus* PropBank.Br (versão 8.0 do Bosque), foram adicionadas ao conjunto de teste. As estatísticas finais dos conjuntos de dados são apresentadas na Tabela 4.2. Algumas observações interessantes sobre estes conjuntos de dados são:

- O *corpus* é desbalanceado. Não existe uma distribuição uniforme do número de argumentos anotados para cada possível papel semântico. Isto é uma desvantagem para algoritmos de AM que sejam treinados com este *corpus*, dado que poderiam tender a atribuir com maior frequência etiquetas de papéis semânticos para as quais havia maior número de dados de treinamento.

- Existem etiquetas semânticas para as quais o número de argumentos anotados é muito baixo – `A4`, `A5`, `AM-DIR`, `AM-EXT`, `AM-REC` – o que poderia impedir que o algoritmo de AM generalizasse apropriadamente e, como consequência, não atribuísse estes papéis semânticos.

- Existem 22 verbos no *corpus* de teste para os quais não existem instâncias anotadas no *corpus* de treinamento. Isto é positivo, porque permite avaliar a escalabilidade das estratégias de aprendizado.

Tabela 4.2: Estatísticas dos conjuntos de dados de treinamento e teste do *bechmark*.

	Treinamento	Teste	Total
Sentenças	3.164	144	3.308
Tokens	57.744	2.352	60.096
Proposições	5.537	239	5.776
Verbos Diferentes	1.001	164	1.023
Argumentos	12.968	536	13.504
A0	2.934	124	3.058
A1	4.937	211	5.148
A2	1.063	38	1.101
A3	111	2	113
A4	74	1	75
A5	1	0	1
AM-ADV	349	20	369
AM-CAU	155	1	156
AM-DIR	13	2	15
AM-DIS	283	11	294
AM-EXT	80	1	81
AM-LOC	751	27	778
AM-MNR	392	18	410
AM-NEG	316	19	335
AM-PNC	166	5	171
AM-PRD	186	6	192
AM-REC	60	5	65
AM-TMP	1.097	45	1.142

4.2 Avaliação

Os STs empregam as três métricas de avaliação padrão: precisão, cobertura e F_1. Como os dados de entrada no nosso *bechmarch* seguem o formato das STs, e espera-se que a saída dos sistemas que o empreguem também siga o mesmo modelo, o *script* oficial de avaliação das STs, *srl-eval.pl*[2], também faz parte do *benchmark*.

As regras de avaliação das STs também são aplicadas. Assim, para que um argumento seja correto, todas as palavras que o compõem, assim como o seu papel semântico, devem ser corretas. Além disso, o argumento verbal de cada proposição é excluído da avaliação. Isto porque, na maioria das vezes, o verbo corresponde ao verbo evocador da proposição (que é um dado de entrada) e é fácil de ser identificado. Então, avaliar o seu reconhecimento superestima o desempenho global do sistema.

[2]Disponível em http://www.lsi.upc.edu/~srlconll/soft.html

4.3 Sistema *Baseline*

O *benchmark* deve possuir um sistema base com o qual outros sistemas possam ser comparados. O sistema *baseline* usado nas STs emprega umas poucas regras simples de anotação, as quais foram adaptadas para o português do Brasil (considerando as etiquetas semânticas e sintáticas do PropBank.Br) para implementar o nosso próprio *baseline* (ver Tabela 4.3). Uma linguista[3] supervisou esta adaptação.

Tabela 4.3: Regras do sistema *baseline*.

1. Anotar o verbo alvo como `V`.
2. Anotar `não` na oração do verbo alvo como `AM-NEG`.
3. Anotar o primeiro `NP` antes do verbo alvo como `A0`.
4. Anotar o primeiro `NP` depois do verbo alvo como `A1`.
5. Anotar o `que` antes do verbo alvo como `A0`.
6. Trocar `A0` e `A1` se o verbo alvo é parte de um `VP` em voz passiva. Um `VP` é considerado em voz passiva se contém os verbos `ser` ou `estar` e o verbo alvo tem a anotação sintática `V-PCP`.

Uma das regras originais (que diz respeito a verbos modais) não foi adaptada porque na versão atual do PropBank.Br não existe anotação sintática apropriada. O desempenho global do *baseline* (Tabela 4.4) não é muito alto, principalmente porque só foram criadas umas poucas regras para três papéis semânticos em particular e só um tipo de alternância sintática. Contudo, estas regras mostraram-se bastante eficazes na anotação do papel semântico `AM-NEG`.

A ideia de ter um sistema *baseline* é fornecer um mecanismo básico de comparação, que permita validar que as soluções criadas não sejam triviais. O objetivo, então, não é obter o melhor desempenho nos dados de teste. Além disso, considerando que `A0` e `A1` são as etiquetas mais comuns no *corpus*, esperar-se-ia que o desempenho de um sistema esteja fortemente influenciado pelo reconhecimento destes papéis semânticos. Assim, achou-se desnecessário criar mais regras específicas para outros papéis.

[3] A autora do PropBank.Br, Magali Sanches Duran.

Tabela 4.4: Desempenho do sistema *baseline* considerando todas as proposições (conjuntos de treinamento e teste) e só aquelas no conjunto de teste. Os resultados globais consideram todos os papéis semânticos no *corpus*.

	Precisão		Cobertura		F_1	
	Todos	Teste	Todos	Teste	Todos	Teste
Global	64,3%	64,6%	39,1%	40,9%	48,6	50,1
`A0`	51,6%	49,7%	72,2%	70,9%	60,2	58,5
`A1`	77,9%	79,4%	53,8%	53,1%	63,6	63,6
`AM-NEG`	79,6%	90,5%	89,6%	100,0%	84,3	95,0

4.4 Um Sistema Supervisionado

Uma hipótese sob a qual está baseado este projeto é: *"os poucos dados anotados disponíveis no corpus PropBank.Br não permitiriam treinar, apropriadamente, um sistema de APS supervisionado"*. Pela revisão bibliográfica realizada, é válido acreditar que essa afirmação seja verdadeira. Porém, é importante obter resultados empíricos que suportem esta afirmação para o português do Brasil. Assim, nesta seção é apresentado um sistema supervisionado implementado usando os recursos disponibilizados pelo *benchmark* descrito previamente.

4.4.1 Estratégia de Anotação

Para um verbo dado, todos os constituintes da sentença são candidatos a argumentos, mas só a um pequeno subconjunto deles o verbo realmente atribui um papel semântico. Com base nesta consideração, uma estratégia de quatro etapas é adotada, com a intenção de reduzir o número de instâncias negativas (constituintes marcados como `NULL`) nas etapas de treinamento:

1. **Identificação do Verbo:** Usa-se a informação da coluna 9 (ver Tabela 4.1) para identificar o verbo alvo da proposição.

2. **Poda:** Usa-se o método de Xue e Palmer (2004) para filtrar os constituintes que claramente não são argumentos semânticos do verbo alvo. Este é um algoritmo recursivo que começa no verbo alvo. No início, retorna os irmãos do verbo como candidatos; depois, move-se ao pai do verbo, e coleta seus irmãos novamente. O processo continua até atingir o nó raiz. Adicionalmente, se um constituinte é um sintagma preposicional (`PP`), seus filhos também são coletados. Por exemplo, na

Fig 4.5, para o verbo alvo receber, a saída do método será: [Ele]$_{\texttt{NP}}$, [o valor a a vista]$_{\texttt{NP}}$, [após 30 dias]$_{\texttt{PP}}$ e [30 dias]$_{\texttt{NP}}$.

[Ele]$_{\texttt{A0}}$ **receberá** [o valor à vista]$_{\texttt{A1}}$ [após 30 dias]$_{\texttt{AM-TMP}}$.

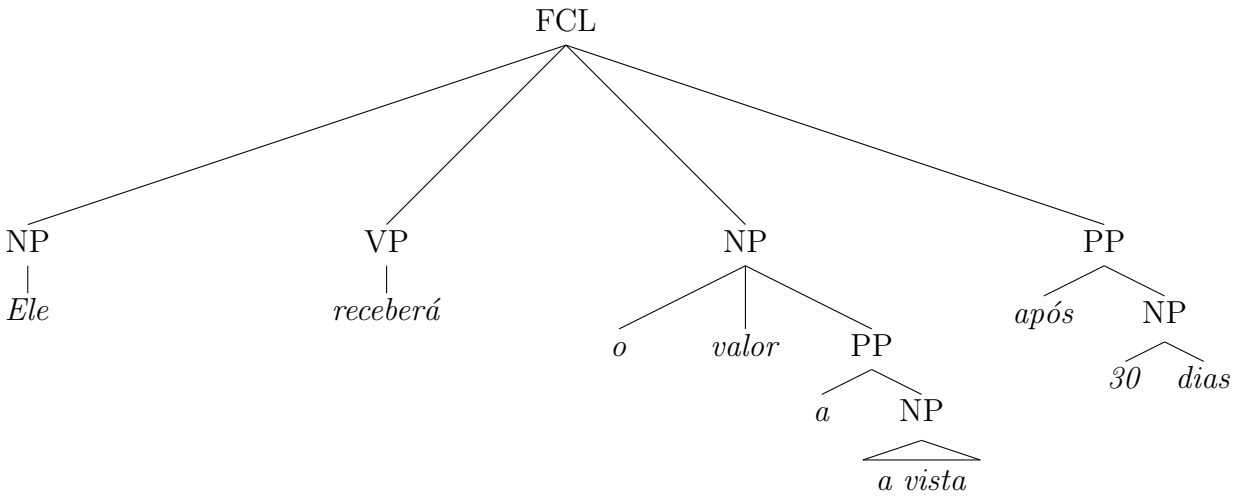


Figura 4.5: Árvore sintática para uma sentença nos dados de treinamento e teste.

3. **Identificação de Argumentos:** Para esta etapa, treina-se um classificador binário para identificar se um candidato é um argumento ou não. Este subsistema recebe como entrada a saída do algoritmo de poda.

4. **Classificação de Argumentos:** Nesta fase, o sistema atribui etiquetas aos candidatos a argumento identificados na etapa anterior. Um classificador multiclasse é treinado para predizer os papéis semânticos dos candidatos. O classificador também pode anotar um candidato como `NULL` (não é argumento) para descartá-lo.

4.4.2 Atributos

Decidiu-se utilizar um subconjunto dos atributos propostos por vários sistemas de APS automática (Gildea e Jurafsky, 2002; Surdenau et al., 2007; Pradhan et al., 2008; Toutanova et al., 2008; Punyakanok et al., 2008; Morante e Bosch, 2009) que são referência na área. Foram necessárias algumas adaptações considerando a anotação sintática disponível nas sentenças dos conjuntos de dados do *benchmark*. O conjunto final de atributos consiste dos seguintes:

- **Caminho:** caminho através da árvore sintática desde o verbo alvo até o constituinte a ser anotado. Por exemplo, na Fig 4.5, o caminho desde receberá até Ele é `VP`↑`FCL`↓`NP`, onde ↑ e ↓ representam subir ou descer na árvore, respectivamente. Descreve a relação sintática entre o constituinte em análise e o verbo alvo.

- **Caminho Parcial: Caminho** na árvore sintática desde o constituinte em análise até o seu menor antepassado comum com o verbo alvo. Tenta generalizar o atributo **Caminho**.

- **Contexto do Predicado:** uma palavra antes e uma depois do verbo alvo, assim como suas etiquetas de *part-of-speech*. Ajuda a capturar variações no sentido do verbo alvo na sentença.

- **Distância em Constituintes na Árvore:** número de constituintes encontrados no **Caminho** desde o verbo alvo até o constituinte a ser anotado. Um constituinte perto ao verbo é mais provável de possuir um papel semântico do que um distante.

- **NEG:** indica se existe um indicador de negação na oração do verbo alvo, usando a estratégia da Regra 2 do *baseline*.

- **Núcleo, Lema do Núcleo, POS do Núcleo:** do constituinte a ser anotado. Os núcleos de sintagmas nominais podem expressar restrições de seleção no tipo de etiqueta de papel semântico que o constituinte em análise pode ser atribuído.

- **Núcleo do Sintagma Preposicional:** se o constituinte é um `PP`, considerar o núcleo do primeiro `NP` dentro dele. Os núcleos dos `PPs` são, geralmente, preposições que não ajudam a discriminar o papel semântico do sintagma. Por exemplo, na cidade e em poucos minutos possuem o mesmo núcleo em, mas o primeiro é `AM-LOC` e o segundo `AM-TMP`. Já os núcleos dos seus `NPs`, cidade e minutos, são mais úteis na distinção de qual etiqueta semântica atribuir.

- **Número de Orações:** quantidade de orações (`FCLs`, `ICLs` e `ACLs`) no **Caminho**, e número de orações na parte ascendente e descendente do **Caminho**. A profundidade do constituinte na árvore sintática indicaria se é realmente argumento do verbo alvo.

- **Número de Sintagmas Verbais:** quantidade de `VPs` no **Caminho**, e número de `VPs` na parte ascendente e descendente do **Caminho**. O comprimento da cadeia verbal indicaria se o constituinte em análise é efetivamente argumento do verbo alvo.

- **Palavras do Constituinte:** a forma, lema e etiqueta POS para as três primeiras palavras que formam o constituinte. Tenta capturar informação lexical e sintática dos *tokens* do constituinte.

- **Parentes do Constituinte:** atributos que indicam o **Tipo de Sintagma**, **Núcleo** e **POS do Núcleo** para o pai, irmão esquerdo e irmão direito do constituinte a ser anotado. Tenta capturar informação lexical e sintática do contexto do constituinte.

- **Posição:** se o constituinte está antes ou depois do verbo alvo. Espera-se alta cooperação com os atributos **Voz** e **Tipo de Sintagma**. Por exemplo, sintagmas nominais na voz ativa que aparecem antes do verbo alvo poderiam possuir o papel semântico `A0`, mas se aparecem depois poderiam ser atribuídos `A1`.

- **Pontuação:** sinal de pontuação à esquerda e à direita do constituinte a ser anotado, ou `NULL` se não existe. É especialmente útil para alguns argumentos adjuntos, como os atribuídos a advérbios que podem aparecer no texto entre vírgulas.

- **Predicado:** forma, lema e etiqueta *part-of-speech* do verbo alvo. Como os papéis semânticos `A0-A5` são específicos para cada verbo, este atributo indicaria ao classificador qual é o verbo alvo da sentença à qual o constituinte em análise pertence.

- **Primeira e Última Palavra/POS no Constituinte:** a primeira e última palavra no constituinte junto com sua etiqueta de *part-of-speech*.

- **Saco de Palavras:** de substantivos, adjetivos e advérbios no constituinte a ser anotado. Tenta capturar informação lexical dos *tokens* do constituinte.

- **SE na Oração do Verbo:** indica a presença da partícula `SE` na oração que contém o verbo alvo.

- **Sequência POS:** cadeia formada pelas etiquetas de *part-of-speech* das palavras que formam o constituinte. Tenta capturar informação sintática dos *tokens* do constituinte em análise.

- **Sequência TOP:** corresponde ao lado direito da regra que expande o nó do constituinte a ser anotado. Tenta capturar informação sintática dos *tokens* do constituinte.

- **Subcategorização:** regra da estrutura do sintagma que expande o nó pai do verbo alvo na árvore sintática. Como no *corpus* Bosque não são usados constituintes `VP` tradicionais, mas *chunks* verbais (contêm principalmente verbos auxiliares e principais), decidiu-se não expandir o nó `VP` mas o seu pai. Para o exemplo na Fig. 4.5, a subcategorização do verbo alvo é `FCL`→`NP-VP-NP-PP`. Tenta-se diferenciar entre usos transitivos e não transitivos do verbo alvo.

- **Tipo de Sintagma:** categoria sintática (`NP`, `VP`, etc.) do constituinte em análise. A sintagmas nominais (`NPs`) geralmente são atribuídos papéis semânticos `A0-A5`, enquanto sintagmas preposicionais (`PPs`) possuem papéis semânticos "adjuntos" (`AMs`).

- **Voz:** se a oração do verbo alvo está em voz ativa ou passiva. A mesma estratégia da Regra 6 do *baseline* foi usada. A distinção entre voz ativa e passiva possui um papel

importante na conexão entre papel semântico e função gramatical, dado que objetos direitos de verbos em voz ativa frequentemente correspondem em papel semântico a sujeitos de verbos na voz passiva.

- Combinações de atributos que tentam capturar a forte co-relação entre eles: **Lema do Predicado + Caminho**, **Lema do Predicado + Núcleo**, **Lema do Predicado + Tipo de Sintagma**, e **Voz + Posição**.

Alguns atributos não conseguiram ser implementados devido à anotação disponível nos conjuntos de dados do *benchmark*, como é o caso de **Categoria Principal**. Este atributo visa determinar se um `NP` é sujeito ou objeto do verbo alvo. Para isso, iniciando do nó do `NP`, sobe na árvore sintática até achar um nó `S` ou `VP`. No primeiro caso, o `NP` seria sujeito e no segundo objeto. Porém, os `VPs` no Bosque só contêm verbos, e não existe nenhuma outra anotação sintática disponível para implementar este atributo[4].

Para extrair o atributo **Núcleo** de cada constituinte, foi utilizada, novamente, informação da CoNLL-X ST. Eckhard Bick, o autor de *parser* Palavras, forneceu regras[5] para transformar o *corpus* Bosque do formato Árvores Deitadas no formato plano de colunas da CoNLL. Com base nessa informação, foi elaborado um conjunto de regras para extrair o núcleo dos constituintes (Tabela 4.5).

Tabela 4.5: Regras para identificação dos núcleos dos constituintes.

1. Em um sintagma nominal (`NP`), o núcleo é o substantivo ou o pronome.
2. Em um sintagma adjetival (`AP`), o núcleo é o adjetivo ou o determinante.
3. Em um sintagma adverbial (`ADVP`), o núcleo é o advérbio.
4. Em um sintagma verbal (`VP`), o núcleo é o verbo auxiliar (geralmente, o primeiro).
5. Em um sintagma preposicional (`PP`), o núcleo é a preposição.
6. Em uma oração finita (`FCL`) ou infinita (`ICL`), o primeiro verbo é o núcleo.
7. Em uma oração averbal (`ACL`), uma unidade composta (`CU`), o qualquer outro caso, o primeiro constituinte contém o núcleo.

[4] O formato TigerXML do PropBank.Br fornece informação de sujeito e objeto através das etiquetas `SUBJ` e `ACC` de dependências do Palavras. Porém, no formato CoNLL o *corpus* contém informação puramente de constituintes.

[5] http://ilk.uvt.nl/conll/data/portuguese/README

4.4.3 *Experimentos e Resultados*

Seguindo a metodologia de Punyakanok et al. (2008), o classificador para identificação de argumentos foi treinado com os constituintes que passaram a fase de poda. Estes podem ter uma etiqueta `ARG` ou `NULL` que indica se realmente são argumentos ou não, sem importar a etiqueta de papel semântico. Depois disto, retomam-se os constituintes que passaram a fase de poda, e são anotados automaticamente pelo classificador treinado para identificação de argumentos. Logo, só aqueles constituintes que recebem uma etiqueta `ARG` são usados para treinar o classificador da fase seguinte. Isto é feito para que o anotador da fase de classificação de argumentos seja treinado para atribuir uma etiqueta de papel semântico (`A0, A1, AM-TEMP`, etc.) ou uma etiqueta `NULL` que indique que o constituinte não é um argumento do verbo alvo.

Realizaram-se experimentos com **Regressão Logística** (RL - também conhecida como **Máxima Entropia**) como algoritmo de aprendizado. Sistemas de APS automática têm usado este algoritmo (He e Gildea, 2004, 2007; Zadeh Kaljahi, 2010) obtendo resultados comparáveis com abordagens computacionalmente mais custosas como *Support Vector Machines*. Usa-se a implementação fornecida no pacote *Scikit-learn*[6] (Pedregosa et al., 2011). Esta implementação do algoritmo possui dois parâmetros que devem ser calibrados para um treinamento apropriado: a **penalidade** usada para regularizar e reduzir a complexidade do modelo de aprendizado, e assim evitar *overfitting* (norma `L1` ou `L2`); e o coeficiente **C** que especifica a força desta regularização (menor valor, maior regularização).

Como não existe um conjunto de dados de desenvolvimento com o qual se possa estimar os parâmetros do algoritmo, utilizou-se a funcionalidade *GridSearchCV* do *Scikit-learn*. Dado um conjunto de valores possíveis para cada parâmetro do algoritmo, *GridSearchCV* avalia todas as possíveis combinações de valores (força bruta) e mantém só as melhores (segundo alguma métrica indicada), utilizando *cross-validation* no conjunto de dados de treinamento. Com esta funcionalidade, usando *10-fold cross-validation* e F_1 como medida de avaliação de desempenho, o classificador para identificação de argumentos obteve o seu melhor desempenho ($F_1 = 97,2$) com $penalidade = L2$ e $C = 1$, enquanto o classificador para classificação de argumentos obteve o seu melhor desempenho ($F_1 = 82,1$) com $penalidade = L1$ e $C = 8$.

Seguindo a prática comum, o sistema de SRL é avaliado em três tarefas: **identificação de argumentos** (etiquetar cada nó como sendo um argumento ou não), **classificação de argumentos** (dados os argumentos *gold*, anotar cada um com a correspondente etiqueta de papel semântico) e a tarefa combinada de **identificação + classificação**. O desempenho do sistema supervisionado nestas tarefas é apresentado na Tabela 4.6[7].

[6] http://scikit-learn.org/

[7] Como na tarefa de classificação o sistema recebe argumentos *gold*, não vai deixar de anotar algum

Tabela 4.6: Resultados do sistema supervisionado nos dados de teste.

Tarefa	Precisão	Cobertura	F_1	Acurácia
Identificação	94,9%	94,0%	94,5	–
Classificação	–	–	–	81,7%
Ident. + Class.	80,0%	79,3%	79,7	–

O desempenho na tarefa de identificação de argumentos é muito mais alto do que nas relacionadas com classificação. Isto pode ser consequência de que o problema de identificação de argumentos é binário – só existem duas etiquetas (`ARG-NULL`) – enquanto no problema de classificação têm-se tantas etiquetas quanto os papéis semânticos existentes no *corpus*. Assim, o classificador de identificação tem uma maior quantidade de instâncias anotadas de aprendizado para cada etiqueta que deve atribuir; já o subsistema de classificação deve lidar com o desbalanceamento dos dados de treinamento.

O sistema supervisionado obtém um desempenho superior ao do *baseline* (na tarefa combinada) nas três medidas de avaliação (Tabela 4.7) tanto de forma global como para os três papéis semânticos específicos. Isto indica que a anotação realizada não é trivial, e que os atributos extraídos são úteis ao algoritmo de aprendizado nas tarefas de classificação.

Tabela 4.7: Comparação de resultados do sistema supervisionado de RL com o *baseline* nos dados de teste.

	Precisão		Cobertura		F_1	
	Baseline	RL	Baseline	RL	Baseline	RL
Global	64.6%	**80,0%**	40.9%	**79,3%**	50.1	**79,7**
`A0`	49,7%	**90,8%**	70,9%	**79,8%**	58,5	**85,0**
`A1`	79,4%	**87,6%**	53,1%	**90,1%**	63,6	**88,8**
`AM-NEG`	90,5%	**95,0%**	**100,0%**	**100,0%**	95,0	**97,4**

A Tabela 4.8 apresenta resultados por papel semântico para a tarefa combinada (identificação + classificação). O sistema classifica melhor os papéis `A0`, `A1` e `AM-NEG`. Possivelmente, porque para os dois primeiros existem mais instâncias anotadas nos dados de treinamento, e porque para `AM-NEG` existe um atributo específico que permite classificá-lo (**NEG**). Dos argumentos adjuntos, para `AM-TMP` existem mais instâncias anotadas nos dados de treinamento e, portanto, é o de melhor desempenho entre os `AMs` (com exceção

argumento, ou anotar algum argumento adicional. Assim, só é apresentado o valor de acurácia.

do AM-NEG). Os papéis semânticos com menor desempenho – A3, A4, AM-DIR, AM-EXT e AM-REC – são precisamente aqueles que possuem o menor número de instâncias anotadas nos dados de treinamento (ver Tabela 4.2). Por esse motivo, o algoritmo de aprendizado não consegue generalizar apropriadamente e erra na sua anotação automática.

Tabela 4.8: Resultados por papel semântico do sistema supervisionado nos dados de teste.

Etiqueta	Corretos	Excedentes	Faltantes	Precisão	Cobertura	F_1
Global	425	106	111	80,0%	79,3%	79,7
A0	99	10	25	90,8%	79,8%	85,0
A1	190	27	21	87,6%	90,1%	88,8
A2	26	18	12	59,1%	68,4%	63,4
A3	0	0	2	0,0%	0,0%	0,0
A4	0	1	1	0,0%	0,0%	0,0
AM-ADV	10	1	10	90,9%	50,0%	64,5
AM-CAU	1	2	0	33,3%	100,0%	50,0
AM-DIR	0	0	2	0,0%	0,0%	0,0
AM-DIS	7	6	4	53,9%	63,6%	58,3
AM-EXT	0	0	1	0,0%	0,0%	0,0
AM-LOC	23	13	4	63,9%	85,2%	73,0
AM-MNR	8	8	10	50,0%	44,4%	47,1
AM-NEG	19	1	0	95,0%	100,0%	97,4
AM-PNC	3	0	2	100,0%	60,0%	75,0
AM-PRD	3	4	3	42,9%	50,0%	46,2
AM-REC	0	0	5	0,0%	0,0%	0,0
AM-TMP	36	15	9	70,6%	80,0%	75,0

Considera-se interessante comparar estes resultados com os obtidos por sistemas estado-da-arte para outras línguas. Na Tabela 4.9 apresentam-se os melhores resultados de sistemas de APS para o inglês (IN) e o espanhol (ES), quando são usadas árvores sintáticas *gold* e os conjuntos de dados de treinamento e teste pertencem ao mesmo gênero.

Pela Tabela 4.9, pode-se dizer que o desempenho do sistema supervisionado na tarefa de identificação de argumentos é próximo aos sistemas estado-da-arte. Porém, isto não acontece na classificação de argumentos. Tanto quando o classificador recebe argumentos *gold* ou identificados automaticamente, os resultados são menores ao estado da arte (especialmente quando comparados com o inglês).

Uma explicação para isso é que a estratégia de três fases usada pelo sistema supervisionado é muito simples quando comparada com as empregadas pelos outros. Por exemplo, Toutanova et al. (2008) utilizam um modelo de anotação conjunta, de tal forma que a atribuição de um papel semântico a um determinado constituinte não é feita isoladamente,

Tabela 4.9: Comparação de desempenho do sistema supervisionado (BR) com outros sistemas estado-da-arte.

Sistema – Língua	Identificação	Classificação	Ident. + Class.
Toutanova et al. (2008) – IN	95,0	91,4%	91,2
Pradhan et al. (2008) – IN	96,8	93,0%	91,2
Surdeanu et al. (2008b) – ES	–	–	84,9
Morante e Bosch (2009) – ES	–	–	84,0
Sistema Supervisionado – BR	94,5	81,7%	79,7

mas leva em consideração as etiquetas semânticas dos outros constituintes na mesma sentença. Por outro lado, Surdeanu et al. (2008b) empregam uma estratégia de inferência conjunta que combina a saída de dois sistemas independentes de APS para obter uma melhor anotação final.

Embora o sistema supervisionado implementado não seja comparável em desempenho com os estado-da-arte, constitui-se no primeiro da abordagem de AM disponível para o português do Brasil. Seus resultados não são triviais (Tabela 4.7) e, portanto, fornece uma base para novas pesquisas na área de APS e espera-se que modificações sejam propostas para aprimorar o seus resultados.

4.5 *Uma Abordagem para Seleção de Atributos*

Os atributos usados na implementação do sistema supervisionado foram selecionados da grande quantidade usada pelos sistemas estado-da-arte, considerando que existe uma descrição clara da sua implementação, que existe informação lexical/sintática no *corpus* PropBank.Br que permita sua implementação ou adaptação, e que não são específicos para a língua do sistema que os propõe. Assim, pode-se dizer que esta seleção foi subjetiva.

Além disso, dado que os atributos foram propostos por diferentes sistemas, existe a possibilidade de que, quando usados em conjunto, haja confusão entre eles, dado que alguns podem fornecer informação redundante ou contraditória. Como consequência, o desempenho do sistema poderia ser afetado.

Considerando o anterior, decidiu-se realizar algum tipo de seleção de atributos e, assim, determinar um (menor) melhor conjunto de atributos para ser incorporado no sistema. Este novo conjunto, por ser menor que o original, permitiria que o sistema fosse treinado/testado mais rapidamente, mas sem afetar significativamente o seu desempenho.

Decidiu-se implementar um método iterativo para seleção de atributos com as seguintes considerações:

1. Calcula-se a **importância unitária** de cada atributo, que corresponde ao valor da métrica usada para avaliar o desempenho do sistema quando este emprega unicamente o atributo.

2. Iniciando com um sistema sem nenhum atributo, acrescentam-se os atributos um por um na ordem decrescente da sua importância unitária.

3. Usando *10-fold cross-validation* no *corpus* de treinamento, calcula-se o desempenho do sistema com o novo atributo.

4. Depois de todos os atributos serem acrescentados, analisa-se a variação do desempenho do sistema, e um subconjunto dos atributos é selecionado ou descartado para a próxima iteração.

5. O processo termina quando todos os atributos utilizados aprimorarem o desempenho do sistema quando acrescentados.

Na seguintes seções, este processo de seleção de atributos é testado na tarefa de identificação de argumentos e na tarefa combinada.

4.5.1 Seleção de Atributos para Identificação de Argumentos

Seguindo o processo descrito previamente, primeiro calcula-se a importância unitária dos atributos no *corpus* de treinamento. Para esta fase, usa-se *10-fold cross-validation* e a medida F_1 para avaliar o desempenho do sistema. A Fig. 4.6 apresenta estes valores na ordem decrescente. No Apêndice A, apresenta-se a equivalência entre as abreviaturas de nomes de atributos usadas pelo sistema implementado (em inglês) e os nomes reais (em português) como descritos anteriormente.

O atributo **Caminho** por si só já permite ao sistema obter um desempenho bastante alto ($F_1 = 96.6$) nesta sub-tarefa, seguido do atributo **Lema do Verbo + Caminho** ($F_1 = 88.7$). O atributo de menor importância é **POS do Núcleo** com $F_1 = 65.4$.

O seguinte passo é treinar iterativamente o sistema, acrescentando os atributos segundo a ordem apresentada na Fig. 4.6. A Fig. 4.7 mostra como o desempenho do sistema foi aumentando e diminuindo, ao mesmo tempo que os atributos eram acrescentados ao treinamento do classificador.

O desempenho final (com todos os atributos) desta iteração, usando *10-fold cross-validation*, é de $F_1 = 97.2$. Analisando a variação de F_1 da Fig. 4.7, devem-se selecionar aqueles atributos que aprimoram o desempenho do sistema. Por exemplo, o atributo **Caminho** é selecionado, mas não o atributo **Lema do Verbo + Caminho** porque mantém o desempenho do sistema igual. **Tipo de Sintagma do Irmão Esquerdo** é

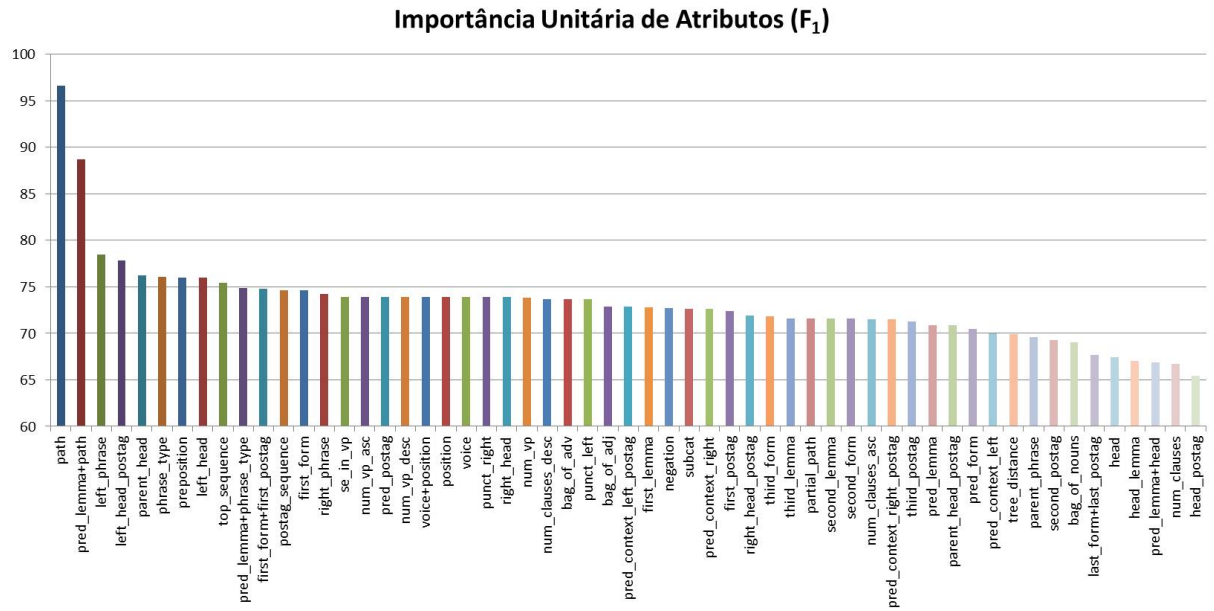


Figura 4.6: Importância de atributos na identificação de argumentos para o sistema supervisionado.

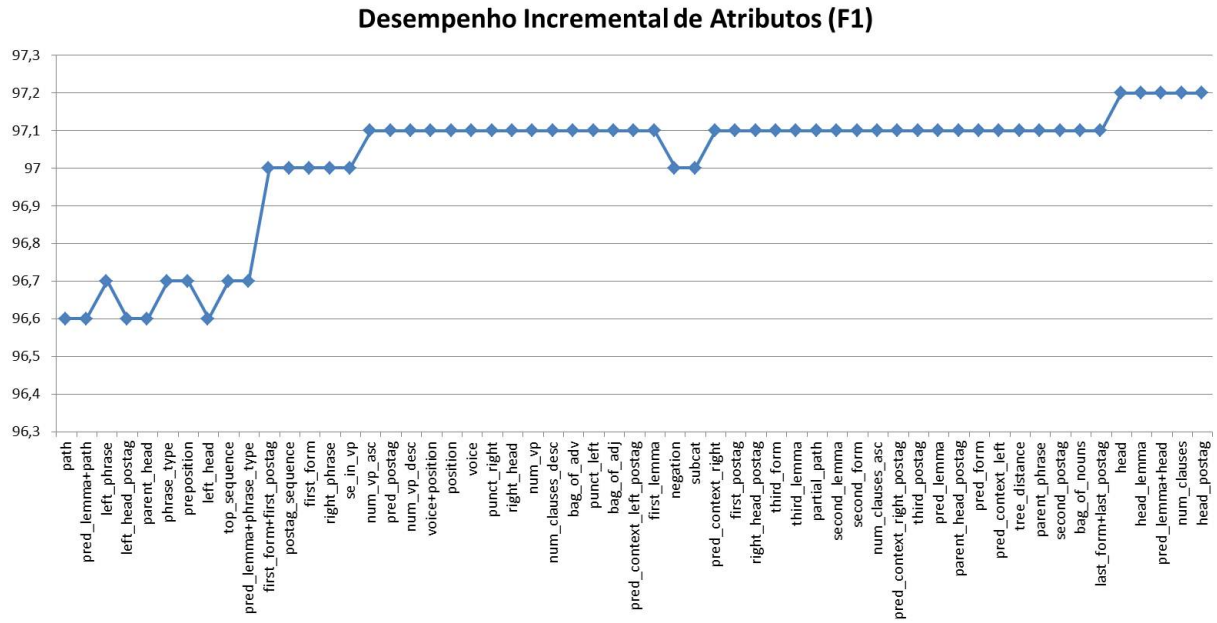


Figura 4.7: Variação inicial do desempenho do sistema supervisionado para identificação de argumentos, quando os atributos são acrescentados iterativamente.

também selecionado, mas não **POS do Núcleo do Irmão Esquerdo** porque decrementa o valor de F_1. O atributo **Tipo de Sintagma** é também selecionado porque melhora o desempenho, embora só consiga atingir um valor alto prévio.

Seguindo este critério, nesta iteração são selecionados os atributos: Caminho, Tipo de Sintagma do Irmão Esquerdo, Tipo de Sintagma, Sequência TOP, Primeira Palavra + POS da Primeira Palavra, Número de Sintagmas Verbais na Parte Ascendente do Caminho, Palavra à Direita do Predicado, e Núcleo.

Com este subconjunto selecionado, realiza-se todo o processo novamente, até que todos os atributos aprimorem o desempenho do sistema quando acrescentados. A Fig. 4.8 apresenta a iteração final, na qual o sistema emprega só 3 atributos: **Caminho**, **Tipo de Sintagma do Irmão Esquerdo** e **Primeira Palavra + POS da Primeira Palavra**, e atinge um desempenho de $F_1 = 97.0$ no conjunto de dados de treinamento. Este valor é levemente menor (0.2 unidades) que o desempenho quando todos os atributos são empregados (Fig. 4.7), porém não é significativo ($p > 0,01$). Portanto, pode-se assumir que os desempenhos são comparáveis.

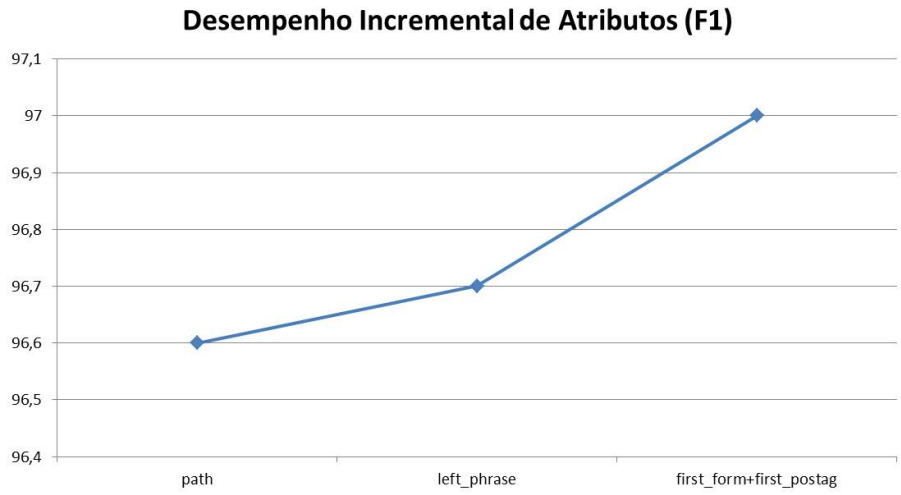


Figura 4.8: Variação final do desempenho do sistema supervisionado para identificação de argumentos, quando os atributos são acrescentados iterativamente.

Finalmente, o classificador de identificação de argumentos com estes 3 atributos foi avaliado no conjunto de dados de teste do *benchmark*. Obteve os seguintes resultados: precisão de 94.4%, cobertura de 93,8%, e F_1 de 94,1. Estes resultados são levemente menores aos obtidos usando o conjunto de atributos completo (Tabela 4.6) mas a diferença não é estatisticamente significativa[8] ($p > 0,01$), o que indica que realmente foram selecionados atributos relevantes para esta tarefa.

[8]Calculado usando SIGF V2 (Padó, 2006)

4.5.2 *Seleção de Atributos para Classificação de Argumentos*

Como explicado anteriormente, este subsistema encarregado da classificação de argumentos é treinado usando os candidatos a argumento automaticamente identificados pela etapa anterior. Para os experimentos aqui apresentados, a etapa de identificação empregou todos os atributos disponíveis.

Segue-se o mesmo processo anterior, gerando primeiro um *ranking* de importância unitária de atributos (Fig. 4.9). Como esperado, os atributos mais importantes para cada sub-tarefa são diferentes. Por exemplo, para classificação de argumentos, o atributo mais importante é **Primeira Palavra + POS da Primeira Palavra** com $F_1 = 61,0$, seguido de **Primeira Palavra** com $F_1 = 56,9$. Já o atributo menos importante individualmente é **Forma do Predicado** com $F_1 = 18,3$.

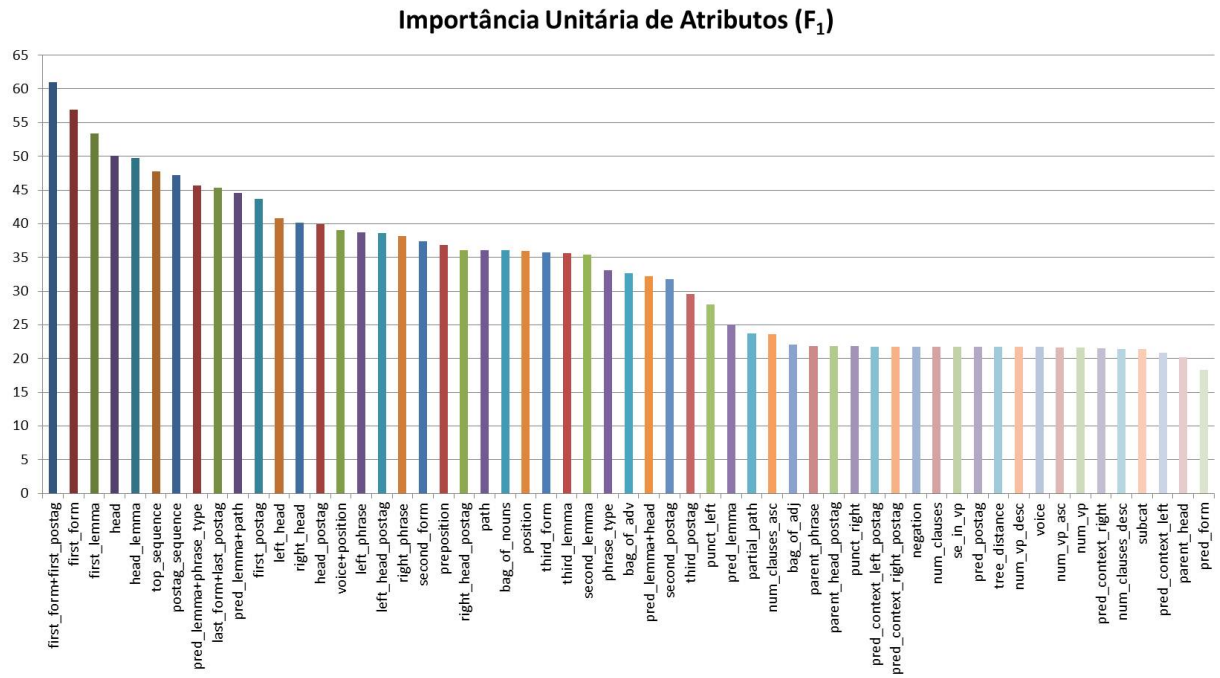


Figura 4.9: Importância de atributos na classificação de argumentos para o sistema supervisionado.

O seguinte passo é analisar a variação do desempenho do sistema (Fig. 4.10) acrescentando os atributos individualmente.

O desempenho final (com todos os atributos) desta iteração, usando *10-fold cross-validation*, é de $F_1 = 81.6$. Da Fig. 4.10, os atributos que aprimoram o desempenho do sistema e devem ser selecionados para próxima iteração são: Primeira Palavra + POS da Primeira Palavra, Forma da Primeira Palavra, Lema da Primeira Palavra, Núcleo, Lema do Núcleo, Sequência TOP, Sequência POS, Lema do Predicado + Tipo de Sintagma,

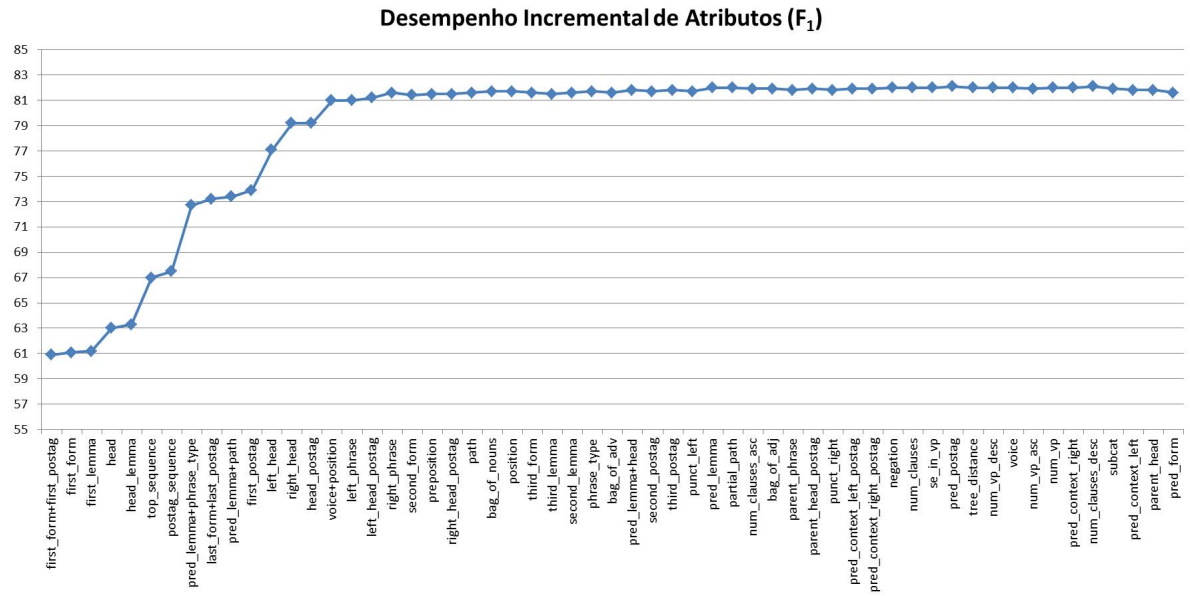


Figura 4.10: Variação inicial do desempenho do sistema supervisionado para classificação de argumentos, quando os atributos são acrescentados iterativamente.

Última Palavra + POS da Última Palavra, Lema do Predicado + Caminho, POS da Primeira Palavra, Núcleo do Irmão Esquerdo, Núcleo do Irmão Direito, Voz + Posição, POS do Núcleo do Irmão Esquerdo, Tipo de Sintagma do Irmão Direito, Núcleo do Sintagma Preposicional, Caminho, Saco de Substantivos, Lema da Segunda Palavra, Tipo de Sintagma, Lema do Predicado + Núcleo, POS da Terceira Palavra, Lema do Predicado, POS do Núcleo do Pai, POS da Palavra à Esquerda do Predicado, NEG, POS do Predicado, Número de Sintagmas Verbais, e Número de Orações na Parte Descendente do Caminho.

Com este subconjunto selecionado, realiza-se todo o processo novamente, até que todos os atributos aprimorem o desempenho do sistema quando acrescentados. A Fig. 4.11 apresenta a iteração final, na qual o sistema emprega só 16 atributos (p.e., **Primeira Palavra + POS da Primeira Palavra**, **Lema da Primeira Palavra**, **Núcleo**, etc.) e atinge um desempenho de $F_1 = 81,6$ no conjunto de dados de treinamento. Este valor é igual ao obtido usando todos os atributos; portanto, aceita-se a seleção realizada.

Da mesma forma que na subtarefa anterior, o subsistema de classificação de argumentos com estes 16 (melhores) atributos é avaliado no conjunto de dados de teste do *benchmark*. Avalia-se para a tarefa de classificação de argumentos e a tarefa combinada, usando o subsistema de identificação de argumentos com todo o conjunto de atributos e só com os selecionados na seção anterior (Tabela 4.10).

A acurácia obtida para classificação de argumentos é igual a quando é usado todo o conjunto de atributos (Tabela 4.6). Na tarefa combinada, usar o subsistema de identi-

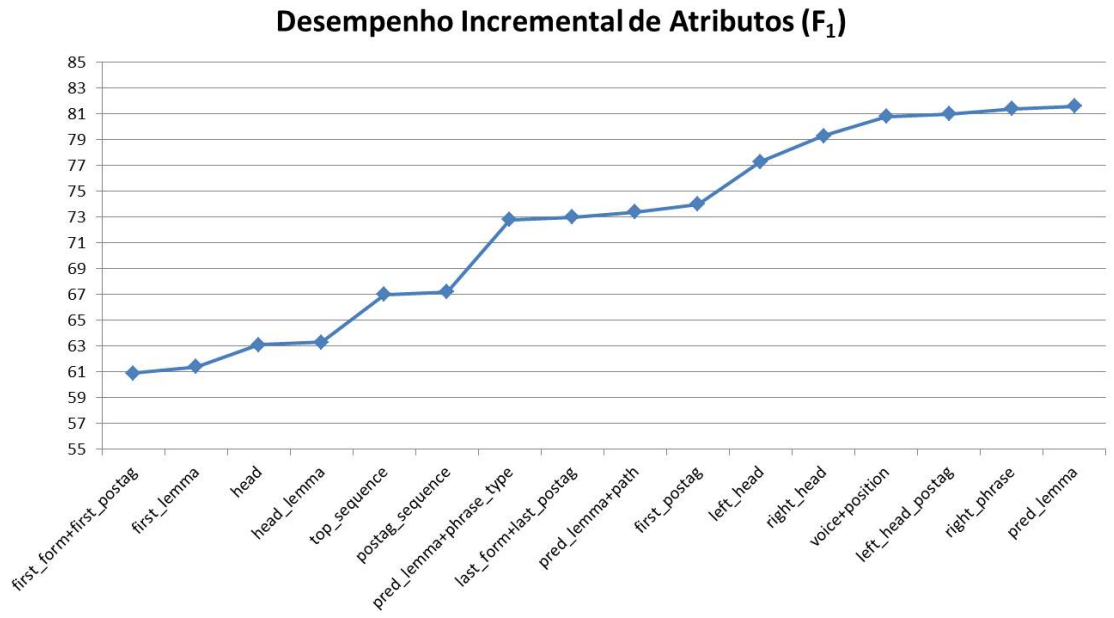


Figura 4.11: Variação final do desempenho do sistema supervisionado para classificação de argumentos, quando os atributos são acrescentados iterativamente.

Tabela 4.10: Resultados do sistema supervisionado com seleção de atributos para classificação de argumentos, com identificação de argumentos usando todos e o subconjunto selecionado de atributos.

	Precisão		Cobertura		F_1		Acurácia
Tarefa	Todos	Selec.	Todos	Selec.	Todos	Selec.	Selec.
Classificação	–	–	–	–	–	–	81,7%
Ident. + Class.	**79,9%**	79,7%	79,1%	**79,3%**	**79,5**	**79,5**	–

ficação com todos os atributos ou só os selecionados produz resultados iguais nos dados de teste ($F_1 = 79,5$). A diferença entre este resultado e o obtido pelo sistema supervisionado original com todos os atributos ($F_1 = 79,7$) não é estatisticamente significativa ($p > 0,01$), o que indica que foram selecionados atributos relevantes para esta tarefa.

4.6 Considerações Finais

Este capítulo é dedicado à anotação automática de papéis semânticos, usando uma abordagem supervisionada.

Apresentou-se um *benchmark* para avaliar o desempenho de sistemas de APS para

o português do Brasil, baseado nas CoNLL *Shared Tasks*. Fornecem-se conjuntos de dados de treinamento e teste, métricas de avaliação, e um sistema *baseline* (simples) baseado em regras. Pretende-se enriquecer este recurso com mais informação sintática (p.e., *chunks*, relações de dependências, árvores sintáticas automáticas) e semântica (p.e., entidades nomeadas). Espera-se que este *benchmark* seja usado para comparar diferentes abordagens na área, o que ajudaria a avançar o estado-da-arte em APS para o português.

Um sistema supervisionado foi desenvolvido usando os recursos fornecidos pelo *benchmark*, com vários objetivos em mente. Em primeiro lugar, demonstrar a utilidade dos recursos fornecidos na implementação de sistemas de APS baseados em *corpus*. Um segundo objetivo é possuir um sistema que sirva como comparação de desempenho do implementado neste trabalho. E em terceiro lugar, verificar se realmente os poucos dados anotados do PropBank não permitem um correto aprendizado supervisionado. Os resultados obtidos para identificação de argumentos são próximos aos de sistemas estado-da-arte para o inglês, mas isso não acontece com a classificação de argumentos. Duas possíveis razões para isso são: que, efetivamente, os dados anotados são insuficientes para generalizar corretamente; ou que os atributos usados não são os mais apropriados.

Finalmente, apresentou-se uma primeira tentativa para seleção de atributos. Esta abordagem resultou ser útil na seleção de atributos permitindo obter desempenhos comparáveis ao do sistema que emprega o conjunto completo de atributos. Contudo, observando os gráficos de variação de desempenho, pode-se perceber que podem ser atingidos valores maiores de desempenho (especialmente, na tarefa combinada). Os atributos selecionados no final do método conseguem atingir um desempenho similar, mas não melhor ao do conjunto completo. Algumas modificações podem ser feitas ao algoritmo de seleção para atingir esta melhora. Por exemplo, ao invés de selecionar um subconjunto dos "melhores" atributos depois de cada iteração, descartar aqueles atributos que diminuem o desempenho do sistema.

Capítulo

5

Abordagem Não Supervisionada: Indução de Papéis Semânticos

Aprender semissupervisionadamente implica empregar técnicas tanto do aprendizado supervisionado quanto do não supervisionado, para aproveitar a informação fornecida por dados anotados e não anotados, respectivamente. No Capítulo 4, estudou-se uma estratégia supervisionada padrão para Anotação de Papéis Semânticos (APS) que extrai atributos dos constituintes das sentenças para treinar um classificador multi-classe encarregado da anotação. Os resultados obtidos pelo sistema de APS implementado evidenciam a importância de possuir bastantes dados anotados que permitam ao classificador generalizar e aprender apropriadamente. Neste capítulo estudam-se estratégias da abordagem não supervisionada, com o objetivo de entender quais características dos constituintes das sentenças poderiam ser aproveitadas para contrabalançar a ausência de uma grande quantidade de dados anotados.

Realizar uma análise de papéis semânticos sem supervisão implica não possuir sentenças com anotações que indiquem verbos, argumentos ou papéis desses argumentos, e não contar com outro tipo de recurso semântico construído manualmente. Como no esquema supervisionado, no não supervisionado o problema é dividido em três tarefas: identificação do verbo, identificação de argumentos e classificação de argumentos. Como no esquema não supervisionado não existe um conjunto de papéis semânticos pré-definido (i.e., não existem instâncias anotadas com as etiquetas semânticas a serem preditas), estes devem ser induzidos dos dados e, portanto, a terceira etapa é chamada de **indução de papéis semânticos** (IPS). Esta estratégia segue a mesma ideia de um problema de *clustering*,

no qual as unidades selecionadas pela etapa de identificação de argumentos são agrupadas em *clusters* que representam um determinado papel semântico.

Neste capítulo são estudados três métodos de IPS adaptados de Lang (2012) para o português do Brasil. Os métodos desenvolvidos induzem um conjunto de *clusters* para cada verbo, i.e., os papéis induzidos são específicos para cada verbo, como no caso do PropBank. Estes métodos assumem que as sentenças para anotação estão sintaticamente analisadas na forma de **árvores de dependências**. Assim, explica-se, primeiro, como foi obtido um *corpus* PropBank.Br com anotação sintática por dependências (Sec. 5.1). Após disso, explica-se como são tratadas as fases de identificação do verbo e dos argumentos, com o objetivo de obter um sistema de APS completo (Sec. 5.2). Como os métodos de IPS não atribuem etiquetas semânticas às instâncias, também são alterados o método de avaliação (Sec. 5.3) e o sistema *baseline* para comparação básica (Sec. 5.4). Após, detalham-se os métodos de IPS, como foram adaptados para o português do Brasil e o resultados obtidos (Sec. 5.5). Finalmente, apresentam-se algumas considerações finais (Sec. 5.6), indicando como os resultados dos experimentos aqui realizados são aproveitados pelo sistema semissupervisionado desenvolvido neste trabalho.

5.1 *O corpus PropBank.Br com Árvores Sintáticas de Dependências*

Os modelos de Lang (2012) visam IPS baseada em árvores sintáticas de dependências. A anotação de papéis semânticos do *corpus* PropBank.Br foi feita sobre os nós das árvores sintáticas de constituintes do *corpus* Bosque. Portanto, realiza-se um processo de transformação no *corpus* que permita obter uma anotação de papéis semânticos baseada em dependências (além de árvores sintáticas de dependências).

O *corpus* PropBank.Br possui o formato TigerXML e as sentenças possuem anotação sintática por constituintes e por dependências. Contudo, as relações de dependência foram estabelecidas entre constituintes e não entre palavras. Assim, para obter uma análise sintática apropriada para os métodos desenvolvidos, é necessário extrair os núcleos de cada constituinte, os quais carregam a relação de dependência sintática. No CoNLL-X *Shared Task* sobre análise sintática de dependências multilíngue (Buchholz e Marsi, 2006), o *corpus* Bosque foi utilizado e transformado do formato Árvores Deitadas[1] ao baseado em colunas usado na CoNLL. Como o *software* usado nesse processo é público, realizaram-se esforços para utilizá-lo, mas não foi possível rodar apropriadamente os *scripts* correspondentes por erros de dependências nas bibliotecas usadas pelo programa. Assim,

[1]Formato plano no qual cada nó da árvore sintática é indentado apropriadamente indicando o nível de profundidade relativo à raiz (`http://www.linguateca.pt/floresta/BibliaFlorestal/`).

implementou-se um *script* para extrair a informação necessária do formato TigerXML do PropBank.Br, usando as mesmas regras empregadas na CoNLL-X (ver Tabela 5.1).

Tabela 5.1: Regras para identificação dos núcleos e dependentes dos constituintes das árvores sintáticas da Floresta Sintá(c)tica.

1. Verbos principais (`MV`) dependem dos verbos auxiliares (`AUX`).
2. Em uma oração, as relações sintáticas de sujeito (`SUBJ`) e subordinador (`SUB`) dependem do verbo finito (`V-FIN`), enquanto as demais dependem do `MV`.
3. Em uma oração finita (`FCL`) ou infinita (`ICL`), o primeiro verbo é o núcleo. Os `SUB`s tornam-se dependentes, mesmo que não possuam uma relação sintática real na oração.
4. Em um sintagma nominal (`NP`), adjetival (`AP`), adverbial (`ADVP`) ou preposicional (`PP`), `H` é núcleo.
5. Em uma oração averbal (`ACL`), o primeiro constituinte é o núcleo (tipicamente, o `SUB`).
6. Em um sintagma verbal (`VP`), o primeiro `AUX` é o núcleo dos constituintes externos à oração, mas para o `MV` dentro do `VP`, o seu núcleo é o último `AUX`.
7. Coordenadores (`CO`) e seguintes elementos conjuntos (`CJT`) dependem do primeiro elemento conjunto.
8. Um par não regular (sem `CJT`s) é tratado como um `ACL`, i.e., o primeiro constituinte é o núcleo se não existe um predicado (`P`). Caso contrario, `X` é o núcleo (se existe algum). Se `X`, por sua vez, é um par regular, isto significa automaticamente que seu `CJT` será o núcleo.

O segundo passo consistiu em transferir a informação de papéis semânticos das árvores de constituintes às de dependências. Para isso, empregou-se o método de Surdeanu et al. (2008a) usado no PropBank, no qual o papel semântico é atribuído ao núcleo do constituinte. As regras para realizar a transferência de informação semântica são apresentadas na Tabela 5.2. A Fig. 5.1 apresenta uma sentença anotada no *corpus* PropBank.Br de dependências e a Tabela 5.3 indica a informação fornecida para cada sentença.

Como descrito, o processo de transformação foi automático. Contudo, para ter certeza de que os dados transformados são confiáveis e possam ser usados nos experimentos com os métodos de IPS, foi realizado um processo de revisão semi-automático:

1. **Automático:** Para cada sentença transformada foi procurada uma igual no *corpus* Bosque usado na CoNLL-X e a anotação sintática foi comparada. Se não existia

Tabela 5.2: Regras para transferência de papéis semânticos.

1. O núcleo de um argumento semântico é atribuído ao *token* dentro dos limites do argumento cujo regente é um *token* fora dos limites do argumento.
2. Se um argumento possui vários núcleos sintáticos, o argumento original é dividido em uma sequência de argumentos descontínuos, i.e., o prefixo `C-` é adicionado à etiqueta de papel semântico.

```
1  Agora          agora          adv      -          9   ADVL   -   -           AM-TMP   -
2  ,              ,              pu       -          8   PU     -   -           -        -
3  os             o              art      M|P        4   >N     -   -           -        -
4  soldados       soldado        n        M|P        8   SUBJ   -   -           A0       -
5  e              e              conj-c   -          4   CO     -   -           -        -
6  a              o              art      F|S        7   >N     -   -           -        -
7  polícia        polícia        n        F|S        4   CJT    -   -           -        -
8  estão          estar          v-fin    PR|3P|IND  0   STA    -   -           -        -
9  trabalhando    trabalhar      v-ger    -          8   MV     Y   trabalhar   -        -
10 juntos         junto          adv      M|P        9   ADVL   -   -           AM-MNR   -
11 para           para           prp      -          9   ADVL   -   -           AM-PNC   -
12 prender        prender        v-inf    -          11  P<     Y   prender     -        -
13 os             o              art      M|P        14  >N     -   -           -        -
14 traficantes    traficante     n        M|P        12  ACC    -   -           -        A1
15 .              .              pu       -          8   PU     -   -           -        -
```

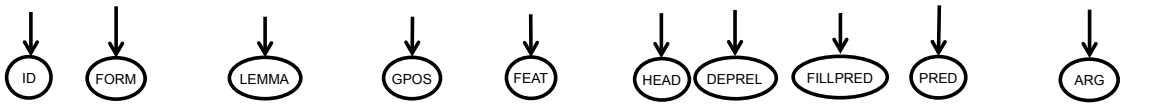


Figura 5.1: Uma sentença anotada no *corpus* PropBank.Br de dependências.

Tabela 5.3: Informação de cada *token* no *corpus* PropBank.Br de dependências.

Número	Nome	Descrição
1	`ID`	Contador de *tokens* que inicia em 1 para cada nova sentença
2	`FORM`	Forma da palavra ou sinal de pontuação
3	`LEMMA`	Lema *gold-standard* da `FORM`
4	`GPOS`	Etiqueta *part-of-speech gold-standard*
5	`FEAT`	Atributos morfológicos *gold-standard*
6	`HEAD`	Regente do *token*, que é ou um `ID` ou zero (0)
7	`DEPREL`	Relação de dependência sintática com o regente
8	`FILLPRED`	Contém Y se o *token* é um predicado alvo
9	`PRED`	Os predicados semânticos na sentença
10...	`ARG`	Colunas com etiquetas de argumentos para cada predicado semântico seguindo a ordem textual

nenhuma diferença, a sentença era considerada como corretamente transformada. Caso contrário, a sentença era separada para ser revista manualmente.

2. **Manual:** Eram analisadas as diferenças entre as sentenças, e comparadas com as regras da Tabela 5.1. Se a diferença era causada por uma implementação errada de alguma regra, esta era corrigida e o processo era iniciado de novo. Já se o erro era devido a diferenças na anotação sintática causada pela diferença de versões dos *corpus* usados[2], a versão transformada era mantida.

Devido a este processo de verificação (especialmente pela revisão manual), esta foi umas das atividades realizadas que envolveu mais tempo. Porém, a qualidade dos dados do *corpus* gerado está garantida.

O PropBank.Br só fornece anotação sintática *gold* (corrigida manualmente) nas suas sentenças. Para obter a anotação automática, seria necessário aplicar algum *parser* – como o Palavras (Bick, 2000) – e realizar todo o processo de transformação novamente. Por uma questão de tempo, isto não foi realizado e, portanto, todos os experimentos realizados neste projeto só foram testados com árvores sintáticas *gold*.

5.2 Identificação do Verbo e dos Argumentos

Como o foco da pesquisa é no problema de IPS, não foi feito maior esforço nas outras etapas (identificação do verbo e dos argumentos). Assim, para elas só foram desenvolvidas regras simples baseadas na informação da árvore sintática.

Para a **identificação do verbo**, no caso de *corpus* PropBank.Br, esta informação é dada nos dados de entrada, como poder ser apreciado na Fig. 5.1 (colunas `FILLPRED` e `PRED`). Portanto, este primeiro problema pode ser considerado resolvido.

No esquema supervisionado, a **identificação de argumentos** indica se um candidato (constituinte ou palavra) é um argumento, i.e., possui algum tipo de papel, embora não indique qual. Porém, neste esquema não supervisionado, esta tarefa consiste em descartar argumentos não semânticos, mas não toma uma decisão final sobre se estes são argumentos ou não. Portanto, vários candidatos que passem esta etapa são certamente argumentos, mas também um pequeno conjunto deles não é. Quando estes candidatos são recebidos pela etapa de IPS, podem ser agrupados em um *cluster* único de não argumentos.

Assim, para identificação de argumentos, as regras desenvolvidas em Lang (2012) para o inglês foram adaptadas para o português do Brasil (Tabela 5.4). Elas permitem descartar ou selecionar candidatos a argumentos e levam em conta as funções gramaticais e as relações sintáticas encontradas na árvore sintática de dependências no percurso do verbo

[2]Na CoNLL-X usou-se a versão 7.3 do *corpus* Bosque, enquanto no PropBank.Br usou-se a versão 8.0.

até o argumento. A priori, todas as palavras na sentença são consideradas candidatos a argumento para um verbo dado. Depois, para cada candidato, as regras são revisadas sequencialmente e a primeira regra que corresponda é aplicada.

Tabela 5.4: Regras para identificação de argumentos para português do Brasil.

1. Descartar o candidato se é um pronome determinativo, conjunção coordenativa ou sinal de pontuação.
2. Descartar o candidato se o caminho de relações desde o verbo alvo até o candidato termina com coordenador, subordinador, etc. (ver Apêndice B para uma lista completa de relações).
3. Manter o candidato se é o sujeito (i.e., regido pela relação sujeito) mais próximo à esquerda do verbo alvo e as relações desde o verbo alvo p até o regente g do candidato são todas para acima (dirigidas como $g \leftarrow p$).
4. Descartar o candidato se o caminho entre o verbo alvo e o candidato, excluindo a última relação, contém as relações sujeito, adjunto adverbial, etc. (ver Apêndice B para uma lista completa de relações).
5. Descartar o candidato se é um verbo auxiliar.
6. Manter o candidato se está diretamente conectado ao verbo alvo.
7. Manter o candidato se o caminho desde o verbo alvo até o candidato conduz por vários nós verbais (cadeia verbal) e termina com uma relação arbitrária.
8. Descartar todos os demais candidatos.

A adaptação das regras foi realizada usando um mapeamento entre as relações de dependência sintática do *corpus* da CoNLL-2008 e do Bosque. Este mapeamento não foi validado por nenhum linguista especialista e, portanto, se for realizada essa verificação é provável que o desempenho deste subsistema melhore.

No *corpus* PropBank.Br de dependências, estas regras obtiveram 74.9% de precisão, 94.4% de cobertura, e 83.5 de F_1. Aqui, precisão mede a porcentagem de argumentos selecionados que são argumentos semânticos reais, enquanto cobertura mede a porcentagem de argumentos semânticos reais que não foram descartados pelas regras.

5.3 Método de Avaliação

Como os modelos de IPS não atribuem um papel semântico real aos candidatos a argumento, não é possível verificar diretamente se a etiqueta é correta comparando-a com

o *gold standard*. Portanto, avalia-se a qualidade do *cluster* como um todo, baseada em quão bem reflete o *gold standard* assumido. Assim, para cada conjunto de *clusters* de um verbo determinado, calculam-se duas medidas: *purity* e *collocation*.

Purity (Manning et al., 2008) é medida como a porcentagem de argumentos que pertencem à classe *gold* majoritária no *cluster* respectivo. No caso de ***Collocation*** (Lang, 2012), para cada papel *gold* é determinado o *cluster* com o maior número de argumentos com esse papel (*cluster* primário desse papel) e depois é calculada a porcentagem de argumentos que pertencem ao *cluster* primário para cada papel *gold*. Seja N o número total de argumentos, G_j o conjunto de argumentos que pertencem à classe *gold* j e C_i o conjunto de argumentos que pertencem ao *cluster* i, estas medidas calculam-se assim:

$$PU = \frac{1}{N} \sum_i \max_j |G_j \cap C_i| \tag{5.1}$$

$$CO = \frac{1}{N} \sum_j \max_i |G_j \cap C_i| \tag{5.2}$$

Finalmente, usa-se a média harmônica de *purity* e *collocation* para obter uma única medida de avaliação da qualidade dos *clusters*.

$$F_1 = \frac{2 \cdot CO \cdot PU}{CO + PU} \tag{5.3}$$

Purity e *collocation* podem ser trivialmente maximizadas colocando, respectivamente, cada instância ou todas as instâncias em um único *cluster*. Logo, sempre devem ser analisadas em conjunto com a F_1, pois uma pode compensar a outra.

5.4 Método Baseline

Pela teoria de *linking* – o mapeamento de papéis semânticos a posições sintáticas – existe uma forte tendência a se relacionar um papel semântico particular a uma função sintática específica como Sujeito, Objeto ou dentro de um Complemento Preposicional usando uma preposição particular. Para validar esta afirmação no *corpus* PropBank.Br, a Tabela 5.5 mostra quão frequentemente papéis semânticos individuais são mapeados a determinadas funções sintáticas, aqui simplesmente definidas como a relação do argumento com seu regente. Como pode ser visto, esta tendência é mantida no *corpus*. Por exemplo, o papel `A0` é geralmente atribuído ao Sujeito (`SUBJ`), `A1` ao Objeto (`ACC`), etc.

Assim, o *baseline* baseia-se na ideia de agrupar os candidatos de acordo com sua função sintática (Algoritmo 1). Para cada verbo, alocam-se tantos *clusters* quantos o número de etiquetas de papel semântico existentes (no caso do PropBank.Br, são 18) mais 1 *cluster default*. Fora do *cluster default*, cada *cluster* é associado com uma função

Tabela 5.5: Tabela de contingência entre função sintática e papéis semânticos. Só as 10 funções sintáticas mais frequentes são apresentadas. Os totais do lado direito incluem as funções sintáticas não apresentadas.

	ADVL	SUBJ	ACC	PIV	SC	SA	PASS	OC	PRED	N<	Total
A0	7	2.775	17	11	4	0	85	0	1	6	2.940
A1	110	1.054	3.338	335	19	31	0	7	0	24	4.946
A2	266	33	99	360	125	71	1	50	1	0	1.043
A3	49	0	11	30	4	3	0	3	0	3	107
A4	18	0	0	11	2	26	0	0	0	0	74
A5	1	0	0	0	0	0	0	0	0	0	1
AM-ADV	340	0	2	0	1	1	0	0	5	0	351
AM-CAU	142	0	1	3	0	0	3	0	5	0	154
AM-DIR	11	0	0	3	0	0	0	0	0	0	15
AM-DIS	267	0	1	0	0	1	0	0	0	0	287
AM-EXT	71	0	4	2	1	0	0	0	0	0	79
AM-LOC	700	0	1	28	0	14	0	0	0	1	750
AM-MNR	359	0	2	9	5	2	0	5	4	0	397
AM-NEG	314	0	0	0	0	0	0	0	0	0	315
AM-PNC	148	0	2	8	2	2	0	1	0	3	168
AM-PRD	121	0	4	2	2	5	2	3	32	2	184
AM-REC	0	1	56	0	0	0	0	0	0	0	63
AM-TMP	1.095	1	4	1	0	2	0	0	4	1	1.115
Total	4.019	3.864	3.542	803	165	158	91	69	52	40	12.989

sintática particular, e todos aqueles candidatos que possuan essa função são mapeadas nesse *cluster*.

Embora o *baseline* seja simples, a seguinte seção demonstrará que é difícil de superar. Isto acontece basicamente porque a grande maioria (aprox. 60%) dos argumentos no PropBank.Br é `A0` ou `A1` e, portanto, o mais importante é a distinção entre estes dois papéis semânticos. Dado que esta pode ser realizada em grande medida na base da função sintática do argumento (como indica a Tabela 5.5), o *baseline* satisfatoriamente reflete este aspecto da tarefa e atinge valores altos de desempenho sem muito esforço.

A Tabela 5.6 apresenta os resultados obtidos quando o método *baseline* é aplicado no *corpus* com duas configurações:

- ***gold*/auto**: árvores sintáticas corrigidas e identificação de argumentos automática usando as regras da Tabela 5.4.

- ***gold*/*gold***: árvores sintáticas corrigidas e candidatos a argumentos verdadeiros, i.e, eles são argumentos mas não se sabe qual é o papel que possuem.

Algoritmo 1: Método *Baseline* de Indução de Papéis Semânticos

Entrada: candidatos a argumentos para um verbo particular
Saída: *clusters* de argumentos específicos para o verbo

$S \leftarrow$ as N posições sintáticas mais frequentes no *corpus*
para cada $s \in S$ **faça**
    alocar um *cluster* c_s para s
fim
alocar o *cluster default* $c_\perp$ para todas as outras posições
para cada *candidato* x **faça**
    $s_x \leftarrow$ posição sintática x
    se $s_x \in S$ **então**
        atribuir candidato ao *cluster* c_{s_x}
    senão
        atribuir candidato ao *cluster default* $c_\perp$
    fim
fim
retorna *todos os clusters*

Como esperado, usar identificação de argumentos *gold* permite obter melhores resultados, como também acontece nos sistemas supervisionados.

Tabela 5.6: Resultados globais do método *baseline*.

	Baseline		
Dados	PU	CO	F_1
gold/auto	73,0	78,5	75,7
gold/gold	75,8	90,1	82,3

Além dos resultados globais, também são apresentados resultados para 10 verbos em particular (Tabela 5.7), que foram selecionados considerando sua frequência (número de proposições) no *corpus*: dizer, fazer, dar, ir, mostrar, falar, informar, fechar, custar e ouvir. Observa-se que a frequência do verbo no *corpus* não afeta o desempenho do *baseline*: verbos com um alto número de proposições ,como dizer, obtêm um desempenho comparável com outros com poucas proposições como informar ou mostrar, e um verbo com ainda menor quantidade de proposições, como custar, obtém o melhor desempenho dentre todos.

Tabela 5.7: Resultados por verbo do método *baseline*.

		Baseline					
		gold/gold			gold/auto		
Verbo	Freq.	PU	CO	F_1	PU	CO	F_1
dizer	252	89,5	95,3	92,3	75,1	89,4	81,6
fazer	167	64,0	85,5	73,2	61,4	70,3	65,5
dar	79	79,3	83,7	81,5	63,2	69,7	66,3
ir	38	51,6	82,4	63,5	52,3	67,6	58,9
mostrar	34	81,2	97,5	88,6	79,0	84,0	81,4
falar	32	63,1	86,2	72,8	58,8	70,6	64,2
informar	21	76,5	90,2	82,8	76,4	87,3	81,5
fechar	12	48,6	77,1	59,6	52,4	73,8	61,3
custar	11	88,0	88,0	88,0	85,2	85,2	85,2
ouvir	7	80,0	100,0	88,9	77,8	83,3	80,5

5.5 *Indução Baseada em Particionamento de Grafos de Similaridade*

Os métodos de Lang (2012) procuram atingir valores de *purity* e F_1 maiores do que os do *baseline* com o objetivo de gerar *clusters* que representem mais adequadamente os papéis semânticos dos candidatos, mas com modelos não triviais, i.e., que mantenham um equilíbrio apropriado entre *purity* e *collocation*.

Estes métodos tentam modelar o fato de dois candidatos a argumento possuírem o mesmo ou diferente papel semântico. Para isso, constrói-se um grafo que conecta os candidatos a argumento, usando um conjunto de funções de similaridade baseadas em atributos sintáticos e lexicais dos candidatos.

5.5.1 *Funções de Similaridade*

Os modelos implementados confiam em julgamentos sobre a similaridade ou dissimilaridade dos papéis semânticos de pares de candidatos a argumentos. Considerem-se as seguintes sentenças:

a. João comeu [o sanduíche].
b. [O sanduíche] foi comido. (5.1)

Os argumentos marcados possuem o mesmo papel semântico, o que pode ser inferido pela sua semântica devido a que o papel de sanduíche não é ambíguo no contexto do verbo comer. O raciocínio aqui é que para um verbo alvo em particular, uma palavra

de conteúdo dada é comumente associada com um único papel semântico. Geralmente, se argumentos de um mesmo verbo coincidem lexicalmente, seus papéis semânticos são susceptíveis de serem os mesmos.

Outro caso a considerar é quando dois argumentos pertencem à mesma sentença (5.2). Aqui, pode-se afirmar que os papéis de cada argumento são diferentes baseados no critério simples de que argumentos que ocorrem na mesma oração (ou, de forma geral, no mesmo *frame*) muito provavelmente não possuem o mesmo papel semântico.

João quebrou [a janela] [com a pedra]. (5.2)

Julgamentos de similaridade podem também estar baseados nas etiquetas *part-of-speech* dos argumentos, embora com menor confiança. Como no caso do critério de *frames*, diferentes etiquetas de *part-of-speech* fornecem evidência negativa, i.e., indicam que os papéis não são iguais. Pelo contrário, evidência positiva é fornecida quando os argumentos possuem a mesma função sintática.

Estes quatro tipos de similaridade baseados no núcleos dos candidatos a argumentos, etiquetas de *part-of-speech*, funções sintáticas e restrições de *frames*, informam os modelos de IPS implementados. A similaridade para cada atributo f é calculada usando uma função $s_f(v_i, v_j)$ que atribui um valor entre $[-1, 1]$ para qualquer par de candidatos (v_i, v_j). Valores positivos de similaridade indicam uma alta probabilidade de que os candidatos possuam o mesmo papel semântico. Valores negativos indicam uma alta probabilidade de que o papel semântico seja distinto. Um valor de zero indica que não existe evidência suficiente para tomar uma decisão.

Os métodos de IPS dependem fundamentalmente das funções de similaridade, seja entre candidatos a argumento ou entre *clusters*. Para calcular as similaridades sintática (s_{syn}), de *frames* (s_{cons}) e de *part-of-speech* (s_{pos}), emprega-se a similaridade de cosseno, representando cada *cluster* como um vetor, cujos componentes são as frequências de um valor particular do atributo para o qual está sendo calculada a similaridade. Para a função de similaridade lexical (s_{lex}) emprega-se um método um pouco mais sofisticado.

Semantic Vectors (Widdows e Cohen, 2010)[3] é um pacote de *software* que cria modelos de espaço de palavras para textos em língua natural. Um *corpus* de referência é tokenizado e indexado usando Apache Lucene[4] para criar uma matriz termo-documento. Após disso, *Semantic Vectors* cria um modelo de espaço de palavras da matriz gerada aplicando projeção aleatória. A classe *CompareTerms* do pacote permite calcular a similaridade entre dois termos, os quais podem ser palavras ou documentos. Esta funcionalidade é usada para criar e comparar os vetores que representam os *clusters* de candidatos durante o cálculo de s_{lex}. Para criar o índice Lucene, foram usados como *corpora* de referência:

[3] http://code.google.com/p/semanticvectors/
[4] http://lucene.apache.org/core/

- ***corpus* PLN-BR FULL.** *corpus* de gênero informativo, subgênero jornalístico, criado durante o projeto PLN-BR[5]. Contém 103.080 mil textos do jornal Folha de São Paulo e 29.014 mil *tokens*.

- ***corpus* Lácio-Ref.** *corpus* aberto e de referência do Projeto Lácio-Web[6], composto de textos em português brasileiro, tendo como característica serem escritos respeitando a norma culta. É um *corpus* cru (não anotado com informações morfossintáticas, sintáticas ou de nível mais elevado), mas possui anotações da existência de elementos gráficos. A grande maioria dos textos está disponibilizada na íntegra.

5.5.2 Representação em Grafos

Nestes modelos baseados em similaridade, toda a informação de cada candidato a argumento está codificada em valores de similaridade com outros candidatos e, portanto, não é possível representar cada um isoladamente. Assim, uma representação natural deste tipo de relação entre os dados é um grafo, cujos vértices correspondem aos candidatos a argumento e cujas arestas têm um peso equivalente à similaridade entre os candidatos. Logo, a IPS é formulada como um problema de particionamento de grafos, no qual o objetivo é dividir o grafo em *clusters* de vértices que representam papéis semânticos.

Dadas as funções de similaridade para vários atributos e um conjunto de candidatos para um verbo em particular, constrói-se um grafo cujos vértices correspondem aos candidatos e cujas arestas representam relações de similaridade entre os candidatos. Como cada atributo possui sua própria função de similaridade, está também associado com seu próprio conjunto de arestas e, portanto, o grafo consiste de várias camadas de arestas; uma para cada atributo (Fig. 5.2). A camada para um atributo em particular conecta pares de candidatos com uma similaridade diferente de zero para esse atributo, com uma aresta cujo peso quantifica a similaridade entre os candidatos em relação ao atributo.

5.5.3 Métodos de Particionamento de Grafos

O problema de particionamento de grafos consiste em encontrar um conjunto de *clusters* que formam uma partição do conjunto de vértices de tal forma que (idealmente) cada *cluster* contenha argumentos de um único papel semântico, e todos argumentos com um papel semântico particular estejam em um único *cluster*. Os métodos desenvolvidos baseiam-se em dois mecanismos que exploram a informação de similaridade no grafo. O primeiro é **aglomeração**, no qual dois *clusters* que contêm candidatos similares são agrupados em um *cluster* maior. O segundo mecanismo é **propagação**, no qual a informação

[5] http://www.nilc.icmc.usp.br/plnbr/
[6] http://www.nilc.icmc.usp.br/lacioweb/index.htm

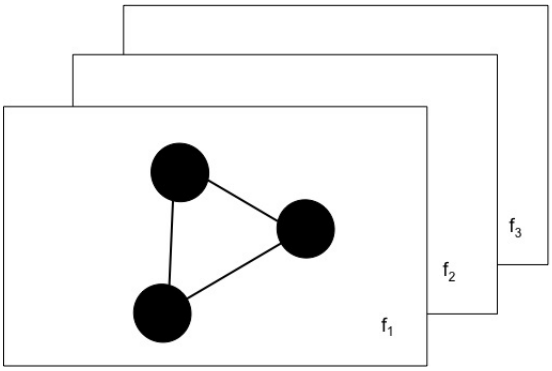


Figura 5.2: Um grafo multicamada no qual cada uma corresponde a um atributo diferente.

da etiqueta do papel semântico é transferida de um *cluster* a outro, baseado na similaridade. Foram desenvolvidos três métodos, um baseado no primeiro mecanismo e dois baseados no segundo.

Particionamento Aglomerativo de Grafos

Este método (inicialmente apresentado em Lang e Lapata (2011a) e estendido em Lang (2012)) iterativamente funde *clusters* de vértices para atingir incrementalmente representações precisas dos papéis semânticos. Depois da iniciação do grafo (descrita posteriormente), o algoritmo começa com um conjunto de *clusters* que possuem alta *purity* mas baixa *collocation*, i.e., os argumentos com um papel semântico particular estão dispersos entre os *clusters*. Depois disso, a *collocation* é iterativamente melhorada executando uma série de fusões de *clusters*, baseadas em uma função que quantifica quão provável é que dois *clusters* possuam argumentos com o mesmo papel semântico. Essencialmente, este processo é apresentado no Algoritmo 2.

A decisão de qual par de *clusters* unir em cada passo é feita pontuando um conjunto de pares de *clusters* candidatos e escolhendo o par com maior pontuação (linha 5). O conjunto de candidatos consiste de pares formados combinando um *cluster* fixo c_i com todos os *clusters* $c_{j'}$ de tamanho maior que c_i.

Embora seja possível iniciar o processo com cada candidato dentro do seu próprio *cluster*, a função de pontuação que é utilizada é mais confiável quando os *clusters* são de maior tamanho. Assim, decide-se obter um conjunto de *clusters* inicial agrupando todos os candidatos que possuem na mesma posição sintática "refinada". Esta considera quatro atributos do candidato: voz verbal (ativa/passiva), posição linear do argumento relativa ao predicado (direita/esquerda), relação sintática do argumento com o seu regente e preposição usada na realização do argumento. Duas posições são iguais se e somente se concordam nos quatro atributos.

Algoritmo 2: Particionamento aglomerativo de grafos para indução de papéis semânticos

```
enquanto não fim faça
    C ← a lista de todos os clusters ordenada descendentemente pelo número de candidatos
    i ← 1
    enquanto i < tamanho(C) faça
        j ← arg max_{0 ≤ j' < i} s(c_i, c_j')
        se s(c_s, c_j) > 0 então
            unir(c_i, c_j)
        senão
            i ← i + 1
        fim
    fim
    atualizar limiares
fim
```

A **função de pontuação** mede a similaridade entre *clusters* e está definida em termos da similaridade entre os candidatos contidos neles. Isto envolve duas etapas de agregação: a primeira sobre as similaridades entre candidatos em cada camada de atributos, resultando em uma pontuação agregada para cada atributo; e a segunda que integra estas pontuações numa única que quantifica a similaridade global entre dois *clusters* (Fig. 5.3).

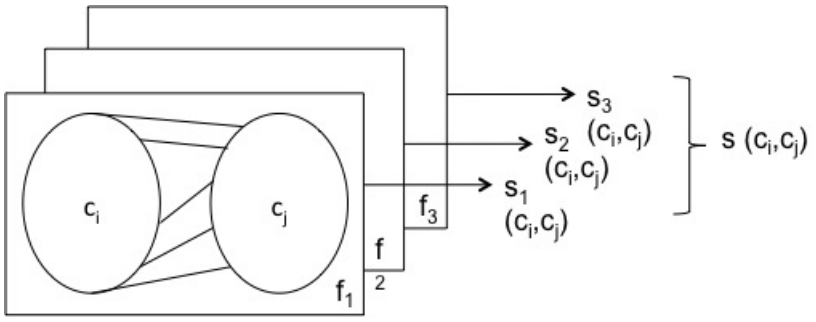


Figura 5.3: Descrição de como a pontuação global de similaridade é calculada entre dois *clusters* para particionamento aglomerativo.

No caso da **agregação por camada** é razoável assumir que um papel semântico em particular impõe uma distribuição específica nos valores dos atributos dos seus argumentos. Assim, é usada a similaridade de cosseno entre *clusters* que reflete similaridade

distributiva:

$$s_f(c_k, c_l) = \frac{x_k^f \cdot x_l^f}{\|x_k^f\| \, \|x_l^f\|} \tag{5.4}$$

A similaridade s_f para um atributo f entre dois *clusters* c_k e c_l é calculada usando as representações vetoriais de cada *cluster* para esse atributo (x_k^f e x_l^f), que contêm como componentes as frequências de ocorrência de um valor particular do atributo.

Como os valores de similaridade e seus agregados para diferentes atributos não são diretamente comparáveis, combinar estes valores através da soma exigiria ponderar a pontuação de cada camada de acordo com sua contribuição. Estes pesos são difíceis de ser calculados e, por isso, no caso da **combinação de pontuações de camadas**, é proposto um método alternativo baseado no tipo de evidência (positiva ou negativa) que cada pontuação agregada por camada fornece:

$$s(c_k, c_l) = \begin{cases} -1 & \text{se } s_{pos}(c_k, c_l) < \alpha, \\ -1 & \text{se } s_{cons}(c_k, c_l) < \beta, \\ s_{lex}(c_k, c_l) & \text{se } s_{lex}(c_k, c_l) > \gamma, \\ 0 & \text{qualquer outro caso.} \end{cases} \tag{5.5}$$

Quando a similaridade de *part-of-speech* (s_{pos}) é inferior a certo limiar α ou quando as restrições de *frame* (s_{cons}) são satisfeitas em menor medida que o limiar β, a pontuação recebe o valor de -1 e a fusão é descartada. Se isto não acontece, o valor da similaridade lexical (s_{lex}) determina a magnitude da pontuação global, desde que seja maior que o limiar γ. Em qualquer outro caso, a função retorna 0 indicando que não existe evidência suficiente para tomar uma decisão positiva ou negativa.

Os parâmetros β e γ são iterativamente ajustados seguindo o procedimento do Algoritmo 3, enquanto α, que determina o valor máximo com o qual uma restrição de *frames* pode ser violada, é fixo em 0.95. Os parâmetros β e γ têm, inicialmente, o valor máximo de 1, assim descartando todas as fusões com exceção daquelas com confiança máxima. Estes valores são diminuídos até serem iguais (ou menores) a $\varepsilon = 0,025$.

Para os experimentos, todo método é testado, primeiro, no conjunto de dados *gold/gold* e só quando os valores de *purity* e F_1 fossem maiores dos que do *baseline*, são realizados testes nos outros conjuntos de dados.

Assim, este primeiro modelo foi testado obtendo os resultados da Tabela 5.8. O valor de *purity* foi incrementado, mas isso não aconteceu com o valor de F_1, principalmente porque a *collocation* foi decrementada de forma considerável.

Realizando uma análise dos *clusters* gerados pelo modelo, determinou-se que existia algum inconveniente com a fórmula de combinação de valores de similaridade, já que

Algoritmo 3: Procedimento de atualização de limiares

```
1 β ← β − 0,025
2 se β ≤ ε então
3 |   β ← 1.0
4 |   γ ← 0.9γ
5 |   se γ < ε então
6 |   |   fim ← verdadeiro
7 |   fim
8 fim
```

Tabela 5.8: Resultados globais do método de particionamento aglomerativo original.

	Baseline			Aglomerativo		
Dados	PU	CO	F_1	PU	CO	F_1
gold/gold	75,8	**90,1**	**82,3**	**79,8**	79,7	79,8

não permitia fundir *clusters* que, de acordo com a análise realizada, deveriam ser unidos (especialmente no caso de *clusters* de menor tamanho).

A ideia por trás dos parâmetros α, β e γ é que ajudem a descartar ou permitir fusões de *clusters* de acordo com os valores de similaridade. Argumentos que pertencem ao mesmo *frame* não podem pertencer, geralmente, ao mesmo *cluster* dado que deveriam possuir diferentes papéis semânticos. Assim, só para valores muito baixos de s_{cons} a fusão deveria ser permitida. Portanto, esta similaridade será melhor controlada pelo valor de α que, agora, é fixo em $0,05$. Por outro lado, s_{pos} pode variar iterativamente e, agora, é limitada pelo valor de β. O comportamento de s_{lex} mantém-se como na fórmula original. Com estas modificações, a função para combinar as similaridades fica como segue:

$$s(c_k, c_l) = \begin{cases} -1 & \text{se } s_{pos}(c_k, c_l) < \beta, \\ -1 & \text{se } s_{cons}(c_k, c_l) > \alpha, \\ s_{lex}(c_k, c_l) & \text{se } s_{lex}(c_k, c_l) > \gamma, \\ 0 & \text{qualquer outro caso.} \end{cases} \tag{5.6}$$

Com esta função modificada, foi testado novamente o método de particionamento aglomerativo multi-camada, obtendo os resultados globais da Tabela 5.9.

Esta modificação permite obter valores de *purity* e de F_1 maiores do que os do *baseline*, tanto no caso de candidatos a argumentos *gold* como nos automáticos. O incremento no valor de *purity* é significativo e corresponde ao esperado quando é comparado com os

Tabela 5.9: Resultados globais do método de particionamento aglomerativo modificado.

	Baseline			Aglomerativo		
Dados	PU	CO	F_1	PU	CO	F_1
gold/auto	73,3	**78,4**	75,7	**77,6**	74,5	**76,0**
gold/gold	75,8	**90,1**	82,3	**81,3**	84,4	**82,8**

resultados obtidos por Lang (2012) no inglês. Porém, isso não acontece com os valores de F_1, nos quais a diferença não é muito significativa, porque o valor de *collocation* sofre um decréscimo importante.

Quando são analisados os valores por verbo apresentados nas Tabelas 5.10 e 5.11, percebem-se duas coisas. Em primeiro lugar, como esperado, o desempenho usando identificação de argumentos *gold* foi consistentemente maior que usando o método automático com as regras elaboradas. Mais interessante ainda é que, quando menos proposições de um verbo estão disponíveis no *corpus*, o ganho em desempenho usando o método aglomerativo é maior que o *baseline* (especialmente, quando é usado o método automático de identificação de argumentos).

Tabela 5.10: Resultados por verbo do método de particionamento aglomerativo modificado no conjunto de dados gold/gold.

		Baseline			Aglomerativo		
Verbo	Freq.	PU	CO	F_1	PU	CO	F_1
dizer	252	**89,5**	**95,3**	**92,3**	86,9	91,4	89,1
fazer	167	64,0	**85,5**	73,2	**70,8**	78,2	**74,3**
dar	79	**79,3**	**83,7**	**81,5**	73,9	77,7	75,8
ir	38	51,6	**82,4**	63,5	**57,1**	76,9	**65,6**
mostrar	34	81,2	**97,5**	**88,6**	**85,0**	87,5	86,2
falar	32	63,1	**86,2**	**72,8**	**64,6**	78,5	70,9
informar	21	**76.5**	90,2	82,8	**76,5**	**92,2**	**83,6**
fechar	12	48,6	**77,1**	59,6	**60,0**	65,7	**62,7**
custar	11	88,0	**88,0**	88,0	**96,0**	**88,0**	**91,8**
ouvir	7	80,0	**100,0**	88,9	**93,3**	93,3	**93,3**

Este resultado é promissor porque evidencia que as medidas de similaridade podem ser aproveitadas para diferenciar os papéis semânticos de argumentos para verbos cuja frequência no *corpus* é baixa. Em geral, os resultados obtidos demonstram que o método de particionamento aglomerativo cumpre com o objetivo de gerar *clusters* não triviais que

Tabela 5.11: Resultados por verbo do método de particionamento aglomerativo modificado no conjunto de dados gold/auto.

		Baseline			Aglomerativo		
Verbo	Freq.	PU	CO	F_1	PU	CO	F_1
dizer	252	**75,1**	**89,4**	**81,6**	74,0	73,5	73,8
fazer	167	61,4	**70,3**	**65,5**	**64,0**	62,4	63,2
dar	79	63,2	**69,7**	**66,3**	**67,4**	61,7	64,4
ir	38	52,3	**67,6**	58,9	**54,1**	66,7	**59,7**
mostrar	34	**79,0**	**84,0**	**81,4**	**79,0**	77,0	78,0
falar	32	58,8	**70,6**	64,2	**63,5**	65,9	**64,7**
informar	21	76,4	87,3	81,5	**80,0**	**89,1**	**84,3**
fechar	12	52,4	**73,8**	61,3	**64,3**	64,3	**64,3**
custar	11	85,2	85,2	85,2	**96,3**	88,9	**92,4**
ouvir	7	77,8	**83,3**	80,5	**88,9**	**83,3**	**86,0**

representam papéis semânticos específicos para um verbo alvo determinado.

Particionamento de Grafos por Propagação de Etiquetas

Como indicado em Lang (2012), este método está baseado na ideia de propagar informação de associação a um determinado *cluster* através das arestas de um grafo, que é derivado do grafo multi-camada original que representa os dados. Cada vértice deste grafo derivado, chamado de **grafo de propagação**, recebe uma etiqueta que indica o *cluster* ao qual o vértice pertence atualmente. O algoritmo de propagação, então, procede iterativamente atualizando a etiqueta de cada vértice, baseado nas etiquetas dos vértices vizinhos e refletindo sua similaridade com o vértice que está sendo atualizado (Fig. 5.4). Este método, quando comparado com o particionamento aglomerativo, é menos propenso a realizar decisões ávidas falsas que não podem ser corrigidas posteriormente, especialmente no caso de valores de pontuação menos confiáveis, i.e., para *clusters* pequenos.

O grafo de propagação é derivado do grafo original dos dados, juntando vários vértices do grafo original em um único vértice do grafo de propagação. Assim, cada vértice deste novo grafo representa um conjunto atômico de candidatos do grafo original que é sempre atribuído ao mesmo *cluster*. Os vértices do grafo de propagação correspondem aos *clusters* dos vértices do grafo original que são obtidos agrupando candidatos pela sua posição sintática "refinada", i.e, são idênticos aos *clusters* iniciais do algoritmo aglomerativo descrito previamente. Este método é explicado no Algoritmo 4.

O procedimento de pontuação de etiquetas requerido na linha 5 do algoritmo está baseado na mesma ideia do procedimento de pontuação do algoritmo aglomerativo descrito

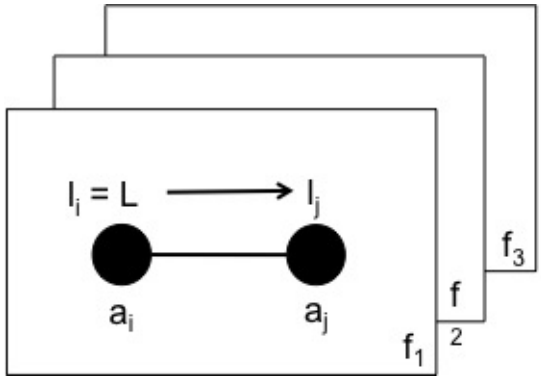


Figura 5.4: Propagação de etiquetas na qual a informação de cada etiqueta de papel semântico é transferida entre os vértices do grafo de propagação.

previamente. Também consiste de duas fases: uma onde evidência é recolhida de forma independente para cada camada de atributos, e a segunda que combina estes valores agregados em um único valor global.

Para explicar a **agregação por camada**, assume-se que o vértice a_i está sendo atualizado. Logo, calcula-se a pontuação s_f para cada atributo f e cada etiqueta l:

$$s_f(l) = \sum_{a_j \in \mathcal{N}_i(l)} s_f(a_i, a_j) \tag{5.7}$$

onde $\mathcal{N}_i(l) = \{a_j | (a_i, a_j) \in B_f, l = l_j, |a_j| > |a_i|\}$ denota o conjunto de vizinhos de a_i com etiqueta l, que possuem um tamanho maior a a_i. Intuitivamente, os vértices vizinhos votam pelo *cluster* ao qual eles pertencem atualmente e a força desse voto é determinada pela similaridade (i.e., peso da aresta) com o vértice que está sendo atualizado.

Para a **combinação de pontuações de camadas** emprega-se uma fórmula similar à usada no algoritmo aglomerativo. Dados os valores de $s_f(l)$ para uma etiqueta particular l em cada camada f, o objetivo é combinar estes valores em um único valor global $s(l)$ para essa etiqueta. Evidência negativa permite rejeitar propagações, enquanto evidência positiva promove uma propagação. Esta função também depende de três parâmetros que podem ser atualizados usando o mesmo método do Algoritmo 3.

$$s(l) = \begin{cases} -1 & \text{se } s_{pos}(l) < \alpha, \\ -1 & \text{se } s_{cons}(l) < \beta, \\ s_{lex}(l) & \text{se } s_{lex}(l) > \gamma, \\ 0 & \text{qualquer outro caso.} \end{cases} \tag{5.8}$$

Experimentos realizados com a versão original do método de propagação multi-camada no conjunto de dados *gold*/*gold*, obtiveram resultados com comportamento similar ao do

Algoritmo 4: Propagação de etiquetas para indução de papéis semânticos

enquanto *não fim* **faça**
  $A \leftarrow$ a lista de todos os vértices de propagação ordenada descendentemente por tamanho (número de candidatos contidos)
  $i \leftarrow 1$
  enquanto $i < tamanho(A)$ **faça**
    $l^* \leftarrow \arg \max_{l \in \{0 \ldots L\}} s(l)$
    se $s(l^*) > 0$ **então**
      $l_i \leftarrow l^*$
    fim
    $i \leftarrow i + 1$
  fim
  atualizar limiares
fim

método aglomerativo original; i.e., os valores de *purity* aumentaram, mas os de *collocation* foram reduzidos de forma tão significativa que a F_1 também diminuiu. Assim, decidiu-se modificar este algoritmo da mesma forma que o método aglomerativo original, modificando como os valores de α, β e γ são usados na fórmula de combinação de pontuações de camadas. Os novos resultados obtidos são apresentados na Tabela 5.12.

Tabela 5.12: Resultados globais do método de propagação de etiquetas modificado.

	Baseline			Propagação		
Dados	PU	CO	F_1	PU	CO	F_1
gold/auto	73,3	**78,4**	75,7	**76,7**	75,0	**75,8**
gold/gold	75,8	**90,1**	82,3	**80,9**	84,8	**82,8**

Novamente, a modificação na fórmula de combinação de pontuações de camadas permite obter valores de *purity* e de F_1 maiores do que os do *baseline*, tanto no caso de candidatos a argumentos *gold* como nos automáticos. Observa-se um comportamento similar aos resultados do algoritmo aglomerativo: (i) o acréscimo no valor de *purity* é significativo, mas não é o caso para os valores de F_1, pelo alto decréscimo no valor de *collocation*; e (ii) quando menos proposições de um verbo estão disponíveis no *corpus*, o ganho em desempenho é maior (Tabelas 5.13 e 5.14).

Em geral, como no caso do algoritmo aglomerativo, os resultados obtidos demons-

tram que o método de propagação de etiquetas cumpre com o objetivo de gerar *clusters* não triviais que representam papéis semânticos específicos para um verbo alvo determinado. Além disso, novamente, as medidas de similaridade ajudam a diferenciar os papéis semânticos de argumentos, especialmente para verbos com baixa frequência no *corpus*.

Tabela 5.13: Resultados por verbo do método de propagação de etiquetas modificado no conjunto de dados gold/gold.

		Baseline			Propagação		
Verbo	Freq.	PU	CO	F_1	PU	CO	F_1
dizer	252	**89,5**	**95,3**	**92,3**	83,3	92,5	90,9
fazer	167	64,0	**85,5**	**73,2**	**69,5**	75,9	72,6
dar	79	**79,3**	**83,7**	**81,5**	73,9	78,8	76,3
ir	38	51,6	**82,4**	63,5	**60,4**	**82,4**	**69,7**
mostrar	34	81,2	**97,5**	**88,6**	**85,0**	88,8	86,8
falar	32	**63,1**	**86,2**	**72,8**	**63,1**	69,2	66,0
informar	21	76.5	90,2	82,8	**80,4**	**90,2**	**85,0**
fechar	12	48,6	**77,1**	59,6	**60,0**	65,7	**62,7**
custar	11	88,0	**88,0**	88,0	**96,0**	**88,0**	**91,8**
ouvir	7	80,0	**100,0**	88,9	**93,3**	93,3	**93,3**

Tabela 5.14: Resultados por verbo do método de propagação de etiquetas modificado no conjunto de dados gold/auto.

		Baseline			Propagação		
Verbo	Freq.	PU	CO	F_1	PU	CO	F_1
dizer	252	**75,1**	**89,4**	**81,6**	72,8	74,5	73,6
fazer	167	61,4	**70,3**	**65,5**	**64,6**	62,0	63,3
dar	79	63,2	**69,7**	66,3	**66,7**	67,4	**67,0**
ir	38	52,3	67,6	58,9	**56,8**	**70,3**	**62,8**
mostrar	34	**79,0**	**84,0**	**81,4**	78,0	81,0	79,5
falar	32	58,8	**70,6**	**64,2**	**61,2**	64,7	62,9
informar	21	76,4	87,3	81,5	**80,0**	**89,1**	**84,3**
fechar	12	52,4	**73,8**	61,3	**61,9**	69,0	**65,3**
custar	11	85,2	85,2	85,2	**96,3**	**88,9**	**92,4**
ouvir	7	77,8	**83,3**	80,5	**88,9**	**83,3**	**86,0**

Combinação Heurística de Similaridades

Os algoritmos de Lang (2012) descritos previamente são inovadores na área porque empregam grafos de várias camadas para representar a similaridade entre os candidatos a argumento. Contudo, é também possível juntar as várias camadas de atributos em um grafo de uma única camada (Fig. 5.5). Assim, o grafo pode ser particionado usando um algoritmo de propagação de etiquetas mais simples, como o apresentado no Algoritmo 5 que é uma versão modificada do Algoritmo 4.

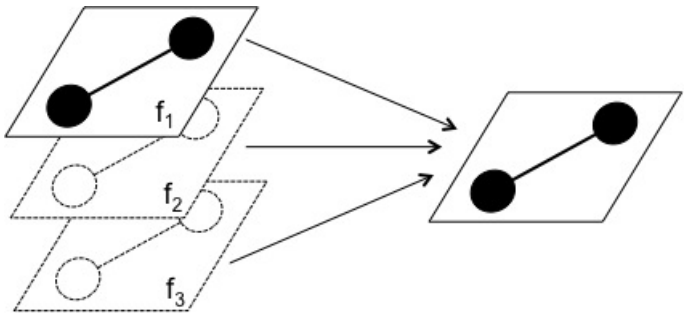


Figura 5.5: Um grafo de uma camada que combina a informação dos atributos heuristicamente.

Nos métodos anteriores, obtém-se uma similaridade agregada para cada camada de atributos e depois calcula-se uma similaridade global **entre *clusters***. Aqui, o método combina as similaridades de atributos em uma pontuação global **entre candidatos**.

Cada vértice v_i do grafo, que aqui representa um candidato em vez de *clusters* atômicos, recebe uma etiqueta $l_i \in \{1 \dots L\}$ que indica o *cluster* ao qual pertence. Como no caso multi-camada, inicialmente, cada vértice pertence ao seu próprio *cluster* e o algoritmo de propagação atualiza a sua etiqueta iterativamente, baseado nas etiquetas dos vértices vizinhos:

$$l_i \leftarrow \arg \max_{l \in \{1 \dots L\}} \sum_{v_j \in \mathcal{N}_i(l)} s(v_i, v_j) \tag{5.9}$$

Aqui $\mathcal{N}_i(l) = \{v_j | (v_i, v_j) \in E, l = l_j\}$ denota o conjunto dos vizinhos de v_i com etiqueta l. Em cada iteração, todos os vértices são atualizados em ordem aleatória. Quando o vértice v_i é atualizado com a etiqueta l, é calculado um valor de confiança da atualização, que é a similaridade média com os vizinhos que possuem etiqueta l:

$$conf(l_i \leftarrow l) = \frac{1}{|\mathcal{N}_i(l)|} \sum_{v_j \in \mathcal{N}_i(l)} s(v_i, v_j) \tag{5.10}$$

Assim, as atualizações com um alto valor de confiança são priorizadas estabelecendo um limiar θ e permitindo só atualizações com uma confiança maior ou igual a θ. Este

Algoritmo 5: Propagação de etiquetas de uma camada para indução de papéis semânticos

enquanto *não fim* **faça**
  $A \leftarrow$ a lista de todos os vértices de propagação em ordem aleatória
  $i \leftarrow 0$
  enquanto $i < tamanho(A)$ **faça**
    $l^* \leftarrow \arg\max_{l \in \{1 \ldots L\}} \sum_{v_j \in \mathcal{N}_i(l)} s(v_i, v_j)$
    $conf \leftarrow \frac{1}{|\mathcal{N}_i(l)|} \sum_{v_j \in \mathcal{N}_i(l)} s(v_i, v_j)$
    se $conf > \theta$ **então**
      $l_i \leftarrow l^*$
    fim
    $i \leftarrow i + 1$
  fim
  atualizar limiares
fim

limiar tem um valor inicial de 1 (máxima confiança) e é diminuído em um valor de 0,0025 depois de cada iteração até atingir um valor mínimo θ_{min}.

Os valores de similaridade de cada atributo são combinados heuristicamente baseados em conhecimento prévio do problema. Isto limita o uso de um pequeno número de similaridades cuja influência relativa com a similaridade global pode ser formulada de forma explícita: s_{lex}, s_{cons} e s_{synt}. Esta última é definida usando os quatro atributos da posição sintática "refinada" empregada na inicialização do grafo multi-camada. Se a relação sintática entre o argumento e o regente não é a mesma, a pontuação é zero. Em qualquer outro caso, a pontuação é $\frac{S}{4}$, com S igual ao número de atributos que têm o mesmo valor.

Com base nessas funções de similaridade para cada atributo, é construída uma função de similaridade global da forma seguinte:

$$s(v_i, v_j) = \begin{cases} -\infty & sse\ s_{cons}(v_i, v_j) = 1, \\ \lambda s_{lex}(v_i, v_j) + (1 - \lambda) s_{syn}(v_i, v_j) & \text{qualquer outro caso.} \end{cases} \tag{5.11}$$

O primeiro caso indica que duas instâncias v_i e v_j que pertencem ao mesmo *frame* não podem possuir o mesmo papel semântico. Formalmente, s possui $range(s) = [-1, 1] \cup \{-\infty\}$ e para $x \in range(s)$ define-se $x + (-\infty) = -\infty$. Quer dizer que quando as pontuações das etiquetas são somadas com algum valor $-\infty$, a pontuação total é $-\infty$, i.e., a propagação é descartada. Para o peso do parâmetro λ do segundo caso, Lang (2012) escolhe o valor de 1/2, baseado na ideia de que, aproximadamente, a similaridade lexical e a sintática possuem igual importância.

A Tabela 5.15 apresenta o desempenho deste modelo no *corpus gold/gold*. Assim como nas versões originais dos modelos anteriores, o valor de *purity* foi aumentado, mas isso não aconteceu com o valor de F_1. A mesma modificação feita nos modelos anteriores não pode ser aplicada aqui, porque este não possui múltiplas camadas. Foram feitos testes mudando o valor de λ, mas os resultados não melhoraram.

Como indicado anteriormente, todo método é testado, primeiro, no conjunto de dados *gold/gold* e só quando os valores de *purity* e F_1 são maiores dos que do *baseline*, são realizados testes nos outros conjuntos de dados. Como nenhuma modificação conseguiu melhorar ambos valores para este modelo, não foram feitos testes usando a identificação automática de argumentos.

Finalmente, a Tabela 5.16 apresenta uma comparação do desempenho (global) dos três modelos implementados nos dois conjuntos de dados utilizados. Em geral, o modelo de particionamento aglomerativo (modificado) obteve o melhor desempenho (como também foi o caso de Lang (2012)), com exceção na *purity* para o conjunto de dados *gold/gold*. Neste último, o modelo de propagação de etiquetas de única camada obtém o valor mais

Tabela 5.15: Resultados globais do método original de propagação de etiquetas de camada única com combinação heurísticas de similaridades.

	Baseline			Aglomerativo		
Dados	PU	CO	F_1	PU	CO	F_1
gold/gold	75,8	**90,1**	**82,3**	**82,7**	71,8	76,9

alto e ainda maior aos obtidos pelas versões originais dos outros modelos. Porém, este resultado deve-se tomar com muito cuidado, porque como o valor de F_1 deste modelo é menor que do *baseline*, não se pode assegurar que os *clusters* gerados sejam não triviais.

Tabela 5.16: Comparação do desempenho dos modelos de indução de papéis nos conjuntos de dados.

	Baseline			Aglomerativo			Propagação			Comb. Heurística		
Dados	PU	CO	F_1	PU	CO	F_1	PU	CO	F_1	PU	CO	F_1
gold/auto	73,3	78,4	75,7	**77,6**	74,5	**76,0**	**76,7**	75,0	**75,8**	–	–	–
gold/gold	75,8	90,1	82,3	**81,3**	84,4	**82,8**	**80,9**	84,8	**82,8**	**82,7**	71,8	76,9

5.6 Considerações Finais

Neste capítulo foram estudados três métodos não supervisionados para indução automática de papéis semânticos propostos por Lang (2012) e foram aplicados com (quase) sucesso para o português do Brasil.

Lang (2012) também realizou experimentos usando outro tipo de medidas de similaridade para os algoritmos de particionamento aglomerativo e de propagação de etiquetas multi-camada. Além disso, para o caso do modelo de camada única, foi realizado um teste com uma quantidade pequena de dados anotados para obter o valor do parâmetro λ e, assim, não utilizar o valor heurístico de 1/2. Porém, o desempenho dos modelos de IPS com estas modificações não foi melhor ao dos algoritmos aqui apresentados. Assim, apesar destas modificações serem interessantes e também terem sido estudadas (embora não descritas na seção anterior), não foram implementadas.

Algumas contribuições para a área de pesquisa são: uma versão do *corpus* PropBank.Br considerando anotação por dependentes, um sistema baseado em regras para identificação de (candidatos a) argumentos de um predicado verbal dentro de um proceso de anotação

de papéis semânticos, um sistema *baseline* não supervisionado de indução de papéis semânticos, e modelos de IPS baseados em particionamento de grafos multi-camada para o português do Brasil.

Uma comparação dos resultados dos modelos de IPS aqui apresentados com os do sistema supervisionado do capítulo anterior (e o semissupervisionado do próximo), não pôde ser realizada por dois motivos:

- **Tipo de anotação sintática.** Os métodos de APS desenvolvidos neste trabalho consideram árvores sintáticas de constituintes. Os atributos usados pelos classificadores foram desenvolvidos considerando este tipo de entrada. Já os métodos de IPS empregam anotação sintática por dependentes. Assim, o tipo de análise realizada e as decisões tomadas pelos modelos dependem deste tipo de anotação. Além disso, a APS atribui papéis semânticos a um conjunto de palavras na sentença (um constituinte), enquanto os métodos de IPS só atribuem papel ao núcleo do argumento. Portanto, a saída de cada sistema é diferente.

- **Papéis semânticos atribuídos.** Os métodos de APS são classificadores que atribuem a cada candidato a argumento uma etiqueta de papel semântico, que corresponde a alguma utilizada no *corpus* PropBank.Br (`A0`, `A1`, `AM-TMP`, `AM-LOC`, etc.). Porém, como os métodos de IPS estão baseados em métodos de *clustering*, não empregam dados anotados e, portanto, as etiquetas atribuídas aos candidatos não correspondem a nenhum papel semântico presente no *corpus*, e só indica o *cluster* ao qual o candidato pertence.

Finalmente, o mais importante dos experimentos realizados neste capítulo é que permitiram estudar que tipo de informação dos argumentos pode ser aproveitada para APS não supervisionada. As medidas de similaridade empregadas mostraram-se úteis para esse objetivo, especialmente no caso de instâncias de verbos com baixa frequência no *corpus*. Um dos maiores problemas para APS para o português é a falta de sentenças anotadas, para uma grande variedade de verbos. Os resultados obtidos pelos métodos de IPS indicam que estes métodos, se forem estudados com maior detalhe, podem colaborar na solução deste problema.

Capítulo

6

Anotação Semissupervisionada com *Self-training*

Um dos maiores problemas para a pesquisa em Anotação de Papéis Semânticos (APS) para o português do Brasil é dispor de poucos dados anotados com este tipo de informação semântica. Isto evita que abordagens tradicionais de aprendizado de máquina supervisionado – usadas com sucesso em outras línguas – sejam empregadas em sistemas de APS para o português e obtenham um desempenho comparável com sistemas estado-da-arte. Como evidenciado no Capítulo 4, uma razão para isso é que o modelo de aprendizado sofre pelo desbalanceamento dos dados, evitando que generalize apropriadamente para todas as possíveis etiquetas de papéis semânticos presentes no *corpus*.

Uma forma de superar o desafio exposto é aproveitar outros (abundantes) dados que, embora não anotados com papéis semânticos, podem fornecer algum outro tipo de informação útil para APS automática. Com este intuito, no Capítulo 5 apresentaram-se modelos não supervisionados de indução de papéis semânticos, os quais demonstraram que existem similaridades sintáticas e lexicais entre argumentos com um mesmo papel semântico que correspondem a um verbo específico, úteis no processo de anotação.

Aproveitando as descobertas realizadas nos experimentos anteriores, neste capítulo descrevem-se diferentes tentativas na implementação de um sistema de APS automática que emprega um algoritmo semissupervisionado simples: *self-training*. As modificações realizadas ao algoritmo tentam lidar com o desbalanceamento do *corpus* de treinamento, e aproveitar a informação fornecida pelas similaridades entre argumentos de um verbo aos quais são atribuídos o mesmo papel semântico.

6.1 *Corpus e Baseline*

Um método de aprendizado semissupervisionado precisa de dados anotados e não anotados; estes últimos em maior número que os primeiros. Como se deseja re-aproveitar os métodos implementados nos capítulos anteriores, decidiu-se usar os dados fornecidos pelo *benchmark* implementado neste trabalho, e dividi-los apropriadamente para cumprir com o requerimento de proporção de tamanho indicado: não anotado >> anotado. Assim, as primeiras 1.000 sentenças do *corpus* de treinamento (aproximadamente, a terça parte) são utilizadas como dados anotados e as restantes como dados não anotados. As estatísticas de cada subcorpus são apresentadas na Tabela 6.1.

Tabela 6.1: Estatísticas dos subconjuntos de dados de treinamento.

	Anotado	Não Anotado	Total
Sentenças	1.000	2.164	3.164
Tokens	18.480	39.264	57.744
Proposições	1.782	3.755	5.537
Verbos Diferentes	583	864	1.001
Argumentos	4.135	8.833	12.968
A0	931	2.003	2.934
A1	1.605	3.332	4.937
A2	325	738	1.063
A3	36	75	111
A4	24	50	74
A5	1	0	1
AM-ADV	94	255	349
AM-CAU	54	101	155
AM-DIR	7	6	13
AM-DIS	87	196	283
AM-EXT	29	51	80
AM-LOC	241	510	751
AM-MNR	112	280	392
AM-NEG	108	208	316
AM-PNC	52	114	166
AM-PRD	62	124	186
AM-REC	13	47	60
AM-TMP	354	743	1.097

A divisão realizada representa um cenário (quase) real, no qual um conjunto de sentenças é escolhido para anotação sem necessariamente considerar o balanceamento dos papéis semânticos dos argumentos dos verbos contidos no *corpus*, mas que reflete o uso real da língua. Quando comparadas as Tabelas 6.1 e 4.2 (estatísticas dos conjuntos de

dados originais), percebe-se que, embora a divisão seja arbitrária (as 1.000 primeiras sentenças consideradas como "anotadas"), a proporção de etiquetas de cada papel no *corpus* de treinamento é mantida. Assim, os papéis semânticos mais frequentes continuam sendo `A0, A1, A2` e `AM-TMP`, e os menos frequentes `A5, AM-DIR` e `AM-REC`.

Nota-se também o número de verbos diferentes no *corpus* de treinamento. Quando comparado com a Tabela 4.2, este diminuiu significativamente (quase em 42%). Isto justifica-se pela distribuição das instâncias anotadas por verbo alvo no *corpus* PropBank.Br. Como apresentado na Figura 1.1, quase 70% dos verbos possui no máximo 4 instâncias anotadas. Logo, a divisão arbitrária só evidencia este outro problema no *corpus*.

Resultados Baseline

O objetivo de implementar um método semissupervisionado é aproveitar os dados não anotados para treinar um sistema de APS de melhor desempenho, do que se fosse treinado supervisionadamente com os poucos dados anotados. Assim, para estabelecer um valor de referência "a ser superado", treina-se o sistema supervisionado do Capítulo 4 usando o subcorpus de treinamento anotado definido previamente. A Tabela 6.2 apresenta os resultados obtidos quando é usado o conjunto de atributos completo.

Tabela 6.2: Resultados globais do sistema supervisionado nos dados de teste quando treinado no subconjunto anotado e com todos os atributos.

Tarefa	Precisão	Cobertura	F_1	Acurácia
Identificação	94,4%	93,7%	94,0	–
Classificação	–	–	–	76,1%
Ident. + Class.	75,2%	74,6%	74,1	–

O desempenho na identificação de argumentos é menor em 0,5 unidades de F_1 ao obtido usando o *corpus* de treinamento completo, mas esta diferença não é estatisticamente significativa ($p > 0,01$). Assim, mesmo tendo um número menor de instâncias de treinamento, estas refletem suficientemente as características dos constituintes que permitem identificá-los como argumentos. Como neste caso o problema de classificação é binário, embora o número de exemplos de treinamento foi reduzido, existem instâncias suficientes de cada possível classe (`ARG, NULL`) para que o classificador aprenda apropriadamente.

Contudo, o decréscimo é maior na tarefa de classificação e na combinada (5,6 unidades menos em acurácia e F_1, respectivamente), com diferença estatisticamente significativa ($p < 0,01$). Isto era esperado devido à redução do número de instâncias anotadas para cada possível papel semântico no *corpus* de treinamento.

Os atributos extraídos pelo sistema supervisionado são todos baseados em informação de constituintes, porque os dados de treinamento do *benchmark* só fornecem esse tipo de informação. No Capítulo 5 foi adicionada informação de dependências às sentenças desse conjunto de dados. Em particular, a função sintática que relaciona cada palavra com o seu regente, e o núcleo *gold* de cada sintagma. Considerando esta nova informação, implementou-se um novo atributo chamado de **Função Sintática** que extrai a relação de dependência do núcleo de um constituinte candidato com o seu regente. Além disso, o atributo **Núcleo** emprega a informação *gold* disponível e não as regras da Tabela 4.5.

Com estas modificações no conjunto de atributos, o sistema supervisionado foi treinado tanto no subconjunto de dados de anotados (Tabela 6.3) como no conjunto completo (Tabela 6.4). O aprimoramento no desempenho do sistema é notório, especialmente nas tarefas relacionadas com classificação de argumentos.

Tabela 6.3: Resultados globais do sistema supervisionado nos dados de teste quando treinado no subconjunto anotado e com atributos de dependências.

Tarefa	Precisão	Cobertura	F_1	Acurácia
Identificação	94,3%	93,3%	93,8	–
Classificação	–	–	–	82.3%
Ident. + Class.	79,8%	78,9%	79,4	–

Comparando os resultados das Tabelas 6.2 e 6.3, observa-se um decremento mínimo na identificação de argumentos (0,2 unidades em F_1), que não é estatisticamente significativo e, portanto, os resultados são comparáveis. Contudo, existe um ganho de 6,2 unidades em acurácia para classificação de argumentos, e de 5,3 unidades em F_1 para a tarefa combinada, o que é estatisticamente significativo ($p < 0,01$).

Tabela 6.4: Resultados globais do sistema supervisionado nos dados de teste quando treinado no conjunto anotado completo e com atributos de dependências.

Tarefa	Precisão	Cobertura	F_1	Acurácia
Identificação	94,9%	93,7%	94,3	–
Classificação	–	–	–	85.5%
Ident. + Class.	83,0%	81,7%	82,3	–

Analisa-se o ganho no desempenho do sistema supervisionado usando todos os dados de treinamento, comparando os resultados das Tabelas 4.6 e 6.4. Novamente, o decréscimo

do valor da F_1 para identificação de argumentos (0,2 unidades) não é estatisticamente significativo. Porém, diferentemente dos resultados para o subconjunto de treinamento, o acréscimo em acurácia (3,8 unidades) para classificação de argumentos, e em F_1 (2,6 unidades) para a tarefa combinada, não é estatisticamente significativo ($p > 0,01$). Isto indica que a informação de dependência sintática mostra-se mais útil quando existem poucos dados de treinamento.

Finalmente, o desempenho do sistema supervisionado no subconjunto de treinamento e o conjunto completo possui uma diferença em F_1 na tarefa combinada de 2,9 unidades, o que não é estatisticamente significativo ($p > 0,01$), mas está no limite ($p \approx 0,03$). Assim, os resultados da Tabela 6.3 representam o ponto de partida para os experimentos semissupervisionados a serem apresentados neste capítulo: o objetivo é aprimorar o desempenho deste classificador supervisionado a fim de que obtenha resultados próximos (ou melhores) aos apresentados na Tabela 6.4, aproveitando a informação dos dados não anotados.

6.2 O algoritmo *Self-training*

O termo *self-training* tem sido usado para se referir a uma variedade de esquemas para usar dados não anotados (He e Gildea, 2007). Aqui adota-se a definição de Clark et al. (2003): *self-training* é um procedimento no qual "um anotador é re-treinado na sua própria *cache* anotada em cada iteração". *Self-training* é um algoritmo de aprendizado semissupervisionado caracterizado pelo fato de que o processo de treinamento utiliza suas próprias predições para se auto-ensinar (Zhu e Goldberg, 2009). O Algoritmo 6 apresenta a forma clássica de *self-training*, sendo a sua ideia básica:

1. Usar um conjunto de dados anotados iniciais para treinar um classificador (`treinar`);

2. Aplicar este classificador a dados não anotados (`etiquetar`) e tomar as predições do classificador como certas para aquelas instâncias com maior confiança (`selecionar`);

3. Expandir os dados anotados, adicionando aqueles etiquetados pelo classificador, e treinar novamente;

4. Repetir este processo – continuar etiquetando novos dados e re-treinando o classificador – até satisfazer uma condição de parada.

No Algoritmo 6, a função `treinar` representa um classificador supervisionado chamado de **classificador base**. Esta é uma das maiores vantagens do *self-training*: é um método

Algoritmo 6: Forma básica do método *self-training*

Entrada: L_0: dados anotados; U: dados não anotados
Saída: c: um classificador treinado

```
c ← treinar(L_0)
repita
    L ← L_0 + selecionar(etiquetar(U, c))
    c ← treinar(L)
até satisfacer condição de parada;
retorna c
```

wrapper[1]. Portanto, a seleção do algoritmo de aprendizado para `treinar` é completamente livre (não limitada a um algoritmo específico).

Sobre a **condição de parada**, Abney (2007) sugere três alternativas para determinar quando terminar o processo:

1. Executar o algoritmo por um número fixo e arbitrário de iterações;

2. Continuar iterando até atingir convergência; i.e., até que os dados anotados e o classificador não mudem mais;

3. Usar *cross-validation* para estimar o número de iterações. Dividir os dados em n partes e alternar cada uma como dados de validação, com as demais partes sendo dados de treinamento. O desempenho nos dados de validação é usado para estimar o número ótimo de iterações; permitindo que cada parte tenha o papel de dados de validação, n diferentes estimativas são obtidas. Calculando a média delas, obtém-se um valor estimado final para o número ótimo T de iterações. Depois todo o conjunto de dados é usado para treinamento, parando logo após T iterações.

Self-training não deve ser confundido com aprendizado incremental. Neste último, todos os dados anotados não estão disponíveis *a priori* e são fornecidos (muitas vezes) um por vez. O método incremental deve ser capaz de incorporar esta nova informação, evoluindo o classificador sem ter que re-treiná-lo completamente. Por sua vez, *self-training* precisa de todos os dados anotados desde um início; se novas instâncias são disponibilizadas, todo o processo iterativo deve ser re-iniciado.

Self-training já foi usado na implementação de sistemas de APS para o inglês. Os resultados obtidos indicam que o algoritmo não necessariamente beneficia o processo de aprendizado: He e Gildea (2007) não conseguiram melhorar o desempenho do classificador supervisionado original, enquanto Lee et al. (2007) e Zadeh Kaljahi (2010) obtiveram ganhos menores, embora estatisticamente significativos. Uma das maiores dificuldades

[1]Utiliza o algoritmo de aprendizado (supervisionado) como uma caixa preta (**?**).

apresentadas é a parametrização dos diferentes componentes do algoritmo, assim como a qualidade dos dados não anotados. Nesse cenário, emprega-se o algoritmo *self-traning* com o fim de iniciar a pesquisa em APS semissupervisionada para o português, mas sem esperar que os resultados obtidos sejam iguais ou superiores ao estado da arte.

6.3 Sistema Semissupervisionado com Self-training

Implementa-se um anotador semissupervisionado de características similares ao sistema supervisionado descrito no Capítulo 4: todo o **conjunto de papéis semânticos** definido no projeto PropBank.Br (`ANs` e `AMs`), a **estratégia** de 3 fases: poda, identificação de argumentos e classificação de argumentos; e todo o conjunto de **atributos** extraídos dos constituintes das sentenças (incluídos os baseados em relações de dependência descritos na Seção 6.1). Adicionalmente, usam-se todos os recursos fornecidos pelo *benchmark* implementado neste trabalho: o ***corpus*** PropBank.Br no formato CoNLL como dados de treinamento e teste (considerando a divisão descrita anteriormente), o *baseline* para comparação básica e a metodologia de avaliação que estima o desempenho do sistema usando precisão, cobertura e F_1.

O **algoritmo** de aprendizado será *self-training*, usando **Regressão Logística** (RL) como classificador base. Os parâmetros da RL são os mesmos estimados para o sistema supervisionado em cada fase da anotação. Para cada instância que deve ser anotada, a RL calcula uma probabilidade para cada etiqueta (papel semântico) possível; e finalmente atribui aquela com maior probabilidade. Esta probabilidade será usada no *self-training* como a **confiança** do classificador supervisionado na anotação.

Seleção de Instâncias Anotadas Automaticamente

A função `selecionar` do algoritmo obtém um subconjunto dos dados etiquetados automaticamente para ser acrescentados aos dados de treinamento. Esta seleção é baseada na confiança do algoritmo supervisionado na anotação. Se é maior (ou igual) a um determinado valor mínimo Ω, a instância deve ser selecionada (Algoritmo 7). Nas primeiras iterações só deveriam ser selecionadas aquelas instâncias com máxima confiança ($\Omega_{max} = 1,0$). Para as seguintes, este valor poderia ser menor. Assim, após cada iteração, Ω seria reduzido em um valor de Δ até um mínimo ε.

Condição de Parada e Confiança Mínima

No algoritmo de *self-training* até agora descrito, falta detalhar dois parâmetros importantes: a **condição de parada**, e a **confiança mínima** ε. Para o caso da condição

Algoritmo 7: Função `selecionar` do algoritmo *self-training*

Entrada: L_{auto}: instâncias anotadas automaticamente com sua confiança
Saída: L_{selec}: instâncias selecionadas

```
para cada (ins, conf) ∈ L_auto faça
    se conf ≥ Ω então
        L_selec ← L_selec + ins
    fim
fim
retorna L_selec
```

de parada, tomando em conta as sugestões de Abney (2007) descritas anteriormente, consideram-se as seguintes possibilidades:

1. Parar quando todas as instâncias não anotadas sejam selecionadas para treinamento do classificador. Para garantir isto, o valor de ε deve ser muito baixo (talvez zero) para assegurar que até as instâncias etiquetadas automaticamente com menor confiança sejam selecionadas. Esta alternativa tem a vantagem de ser simples de implementar, mas possui a desvantagem de poder incorporar dados de baixa qualidade no treinamento do classificador.

2. Parar quando o limiar de confiança Ω atingir o valor mínimo ε. Como no caso anterior, esta abordagem tem a vantagem de ser fácil de implementar, mas possui a desvantagem de precisar estimar um valor de ε que resulte em empregar a maior quantidade de dados não anotados, mas com anotações automáticas confiáveis.

3. Parar quando atingir convergência do classificador ou dos dados anotados. Se depois de n iterações não são selecionadas novas instâncias etiquetadas automaticamente para re-treinar o classificador (i.e, este não muda), o algoritmo termina. Neste caso, convém manter um valor de Ω fixo que não seja muito alto nem baixo, ou controlar o valor ε como na abordagem anterior.

Como a primeira alternativa não garante um re-treinamento confiável, foi descartada e decidiu-se combinar as duas últimas abordagens como condição de parada:

1. O algoritmo *self-training* itera enquanto o classificador for re-treinado;

2. Quando já não foram selecionadas novas instâncias automaticamente etiquetadas, o valor de Ω é decrementando em $\Delta = 0,05$ e se incrementa o contador de iterações;

3. Quando um conjunto de instâncias é selecionado, Ω e o contador de iterações consecutivas são re-iniciados;

4. O algoritmo termina quando o valor de ε é atingido, quando o classificador não é re-treinado após n iterações consecutivas, ou quando já foram etiquetadas todas as instâncias não anotadas.

O valor de Ω é re-iniciado em 3 porque a confiança do classificador deveria aumentar com o re-treinamento, e sempre tenta-se selecionar instâncias etiquetadas com alta confiança. Implementa-se esta condição de parada como apresentado no Algoritmo 8.

Algoritmo 8: Método *self-training* com condição de parada especificada.

Entrada: L_0: dados anotados; U: dados não anotados
Saída: c: um classificador treinado

```
c ← treinar(L_0)
L ← L_0
Ω ← Ω_max
repita
    L_selec ← selecionar(etiquetar(U, c))
    se tamanho(L_selec) > 0 então
        L ← L + L_selec
        U ← U − L_selec
        c ← treinar(L)
        n ← 0
        Ω ← Ω_max
    senão
        Ω ← Ω − Δ
        n ← n + 1
    fim
até n = n_max ou Ω ≤ ε ou tamanho(U) = 0;
retorna c
```

Como no sistema supervisionado descrito no Capítulo 4, usou-se a *GridSearchCV* do *scikit-learn* para estimar os valores de n e ε. Testaram-se valores de $n = [2, 3, 4, 5]$ e $\varepsilon = [0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95]$ usando *10-fold cross-validation* e F_1 como métrica de avaliação. Esses valores foram escolhidos considerando que sempre se deseja selecionar instâncias anotadas com alta confiança.

Com *GridSearchCV*, o subsistema para identificação de argumentos obteve o seu melhor desempenho ($F_1 = 97,2$) com $n = 5$ e $\varepsilon = 0,5$, enquanto o subsistema de classificação obteve o seu melhor desempenho ($F_1 = 79,7$) com $n = 2$ e $\varepsilon = 0,65$. Na Tabela 6.5 apresentam-se os resultados do sistema semissupervisionado nos dados de teste usando esta configuração de parâmetros e a versão de *self-training* do Algoritmo 8.

Para a tarefa de identificação de argumentos, os resultados do *self-training* básico (Tabela 6.5) são minimamente maiores aos do sistema supervisionado treinado no subconjunto anotado (Tabela 6.3). O ganho de maior valor (0,7 unidade) é na cobertura, mas

Tabela 6.5: Resultados globais do sistema semissupervisionado nos dados de teste usando *self-training* básico.

Tarefa	Precisão	Cobertura	F_1	Acurácia
Identificação	94,4%	94,0%	94,2	–
Classificação	–	–	–	83,0%
Ident. + Class.	79,8%	79,5%	79.6	–

como o ganho na precisão (0,1 unidade) não é grande, o incremento na medida F_1 (0,4 unidades) não é estatisticamente significativo ($p > 0,01$). Porém, cumpre-se com o objetivo de aproximar estes resultados aos do supervisionado treinado no conjunto completo de dados (Tabela 6.4). Embora este último tenha uma precisão levemente melhor (0,5 unidade a mais), a cobertura do *self-training* é maior em 0,3 unidades. Como consequência, a diferença em F_1 de 0,1 unidade não é estatisticamente significativa ($p > 0,01$), e ambos desempenhos são comparáveis.

O subsistema de classificação de argumentos obteve resultados (Tabela 6.5) um pouco maiores que o supervisionado (Tabela 6.3) nas duas tarefas em que foi avaliado. No caso da classificação, o incremento foi de 0,7 unidades em acurácia, enquanto que na tarefa combinada foi de 0,2 unidades em F_1, por causa do ganho na cobertura (0,6 unidades). Quando comparados com os resultados objetivo da Tabela 6.4, embora a diferença não seja estatisticamente significativa ($p > 0,01$), a diferença em valor (2,7 unidades em F_1 na tarefa combinada, e 2,5 unidades em acurácia para classificação) ainda não é mínima como no caso da identificação de argumentos. Portanto, aqui não se pode indicar que o objetivo de aproximar os resultados foi cumprido.

6.4 Análise e Aprimoramento do Self-training

Considerando os resultados anteriores, nesta seção apresentam-se diferentes modificações ao funcionamento básico do algoritmo *self-training*, procurando obter resultados melhores aos já apresentados. Os esforços focam-se em melhorar o desempenho do subsistema de classificação de argumentos, cujo aprendizado é o mais afetado pela redução do número de dados anotados para treinamento.

Realiza-se uma análise detalhada do processo de aprendizado do *self-training*, com o objetivo de entender melhor como são aproveitados os dados não anotados pelo algoritmo. Isto permite propor modificações mais apropriadas para tratar os problemas apresentados pelo algoritmo e, assim, melhorar os resultados obtidos até o momento.

Para realizar esta análise, em cada iteração do (re)treinamento foram registrados os seguintes dados para cada candidato: o número da iteração, o papel semântico *gold*, o papel semântico automático, a confiança do classificador, a confiança mínima para seleção (Ω), e se o candidato foi selecionado ou não. Com base nos dados obtidos, implementaram-se diferentes modificações ao algoritmo básico, descritas a seguir.

6.4.1 Condição de Parada Simplificada

O processo de treinamento do subsistema de classificação de argumentos recebe como entrada a saída do subsistema de identificação. Assim, após usar o subsistema de identificação nos dados não anotados, este retorna 8.391 candidatos. Segundo a Tabela 6.1, o subconjunto não anotado possui 8.833 argumentos. Isto quer dizer que, desde o início, o número de instâncias de treinamento é menor ao tamanho que deveria. Convém analisar como o algoritmo *self-training* aproveita estos dados não anotados. Em particular, deseja-se saber se utiliza a maior quantidade possível deles.

O *self-training* do Algoritmo 8, com a qual foram obtidos os últimos resultados na seção anterior, realiza 50 iterações no seu treinamento. Na Tabela 6.6 apresentam-se algumas estatísticas dos candidatos não anotados que sobraram após o término do algoritmo.

Tabela 6.6: Estatísticas dos candidatos não anotados restantes na última iteração de treinamento do sistema semissupervisionado usando *self-training* básico.

Confiança	Corretos	Incorretos	Total
0,95 – 1,00	72	477	549
0,90 – 0,95	66	497	563
0,85 – 0,90	40	297	337
0,80 – 0,85	26	231	257
0,75 – 0,80	20	144	164
0,00 – 0,75	161	998	1.159
Total	385	2.644	3.029

Observa-se que 3.029 candidatos não foram usados no processo de treinamento, o que representa um 36% do total de dados não anotados disponíveis no início. Além disso, existem candidatos para os quais o classificador tinha predito corretamente a sua etiqueta de papel de semântico com uma confiança alta ($> 0,75$); contudo, o algoritmo não os considerou como novas instâncias para o retreinamento.

Analisando os resultados, observou-se que a causa disso é uma inapropriada interação entre dois parâmetros que formam parte de condição de parada do algoritmo: o número

de iterações consecutivas máximo sem modificação do classificador ($n = 2$) e a confiança mínima ($\varepsilon = 0,65$). Duas iterações consecutivas só permitem ao algoritmo considerar instâncias anotadas automaticamente até com confiança mínima de 0,975 e não 0,65. Assim, instâncias que bem poderiam ter beneficiado o treinamento do classificador não são aproveitadas como se deveria.

O objetivo de usar n é controlar a convergência do classificador, para não continuar o seu treinamento se não está sendo modificado. Determinou-se que isto já é controlado pelo valor de Ω. Cada vez que são selecionados novos candidatos (sem importar o número), Ω é reinicializado para sempre tentar obter instâncias anotadas automaticamente com a mais alta confiança. Quando nenhuma nova instância é selecionada, Ω é decrementado para considerar um novo conjunto de instâncias. Como o algoritmo termina quando Ω atinge ε, já se está controlando que o classificador não muda mais, dentro do universo de instâncias automaticamente anotadas com alta confiança.

Portanto, modificou-se a condição de parada (e o algoritmo) para não considerar mais o uso de n. Usou-se *GridSearchCV* novamente para determinar o melhor valor para ε nessa nova configuração. O sistema obteve seu melhor desempenho ($F_1 = 79,5$) com ($\varepsilon = 0,85$. Na Tabela 6.7 apresentam-se os resultados nos dados de teste usando esta modificação na condição de parada do algoritmo *self-training*.

Tabela 6.7: Resultados globais do sistema semissupervisionado nos dados de teste usando *self-training* com condição de parada simplificada.

Tarefa	Precisão	Cobertura	F_1	Acurácia
Identificação	94,4%	94,0%	94,2	–
Classificação	–	–	–	82,7%
Ident. + Class.	80,0%	79,7%	79,8	–

Os resultados indicam uma leve melhora no desempenho do sistema na tarefa combinada (ganho de 0,2 unidades em F_1), causado pelo pequeno acréscimo nos valores de precisão e cobertura. Porém, na tarefa de classificação, a acurácia diminuiu em 0,3 unidades. Embora a diferença nos resultados não é significativa ($p > 0,01$), vale a pena analisar se esta modificação melhorou o aproveitamento dos candidatos não anotados por parte do algoritmo. Agora o algoritmo realiza 685 iterações para treinar e as estatísticas dos candidatos que sobram são apresentadas na Tabela 6.8.

Observa-se que o número de candidatos não anotados que não foram aproveitados pelo algoritmo diminuiu: só ficaram 819 (9,7% dos candidatos disponíveis inicialmente). Além disso, o número de candidatos corretamente preditos com uma confiança alta ($>= 0,75$)

Tabela 6.8: Estatísticas dos candidatos não anotados restantes na última iteração de treinamento do sistema semissupervisionado usando *self-training* com condição de parada simplificada.

Confiança	Corretos	Incorretos	Total
0,95 – 1,00	0	0	0
0,90 – 0,95	0	0	0
0,85 – 0,90	6	18	24
0,80 – 0,85	10	77	87
0,75 – 0,80	10	82	92
0,00 – 0,75	94	522	616
Total	120	699	819

e que não foram usados para o retreinamento é baixo (26). Isto indica que o valor de ε escolhido é realmente apropriado.

Pelos resultados apresentados, pode-se dizer que a modificação implementada permite ao algoritmo aproveitar melhor as instâncias não anotadas que foram etiquetadas automaticamente com alta confiança, mas sem diminuir significativamente seu desempenho. Em realidade, permite aproximá-lo – na tarefa combinada – aos valores objetivo traçados.

6.4.2 Seleção Balanceada

Pretende-se analisar o processo de treinamento com relação a como os candidatos são selecionados, a como é a distribuição dos papéis semânticos destes candidatos e como isto poderia afetar o desempenho do sistema. Usando dados do treinamento do sistema modificado da seção anterior, na Fig. 6.1 apresenta-se a distribuição dos papéis semânticos dos candidatos selecionados para iterações nas quais selecionaram-se mais de 50 instâncias.

O algoritmo seleciona candidatos anotados como `A0` e `A1` em maior quantidade que outros em (quase) todas as iterações. Como no início existe um maior número de dados anotados para argumentos com estes papéis semânticos, o classificador possui maior confiança para atribuí-los e por isso existe essa tendência a possuir um alto número de candidatos selecionados com `A0,A1`.

Nas primeiras iterações, esta tendência ajudaria a incrementar o número de instâncias de treinamento para papéis semânticos de alta frequência. A confiabilidade destas seleções é alta porque foram usados dados *gold* no início. Contudo, dado que nas sucessivas iterações o classificador é treinado usando anotações automáticas, a confiabilidade diminui. Portanto, continuar selecionando um alto número de instâncias anotadas com papéis muito frequentes no início do treinamento, evitaria que o algoritmo de aprendizado generalize.

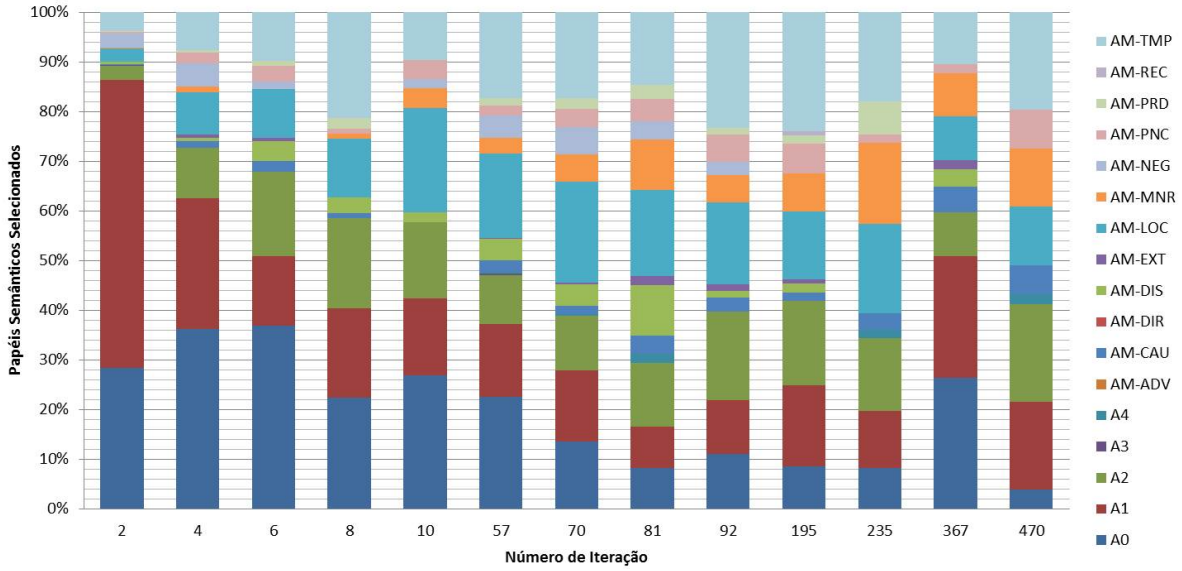


Figura 6.1: Distribuição das etiquetas automáticas de papéis semânticos dos candidatos selecionados em cada iteração.

Assim, outra modificação consiste em evitar que um alto número de instâncias anotadas automaticamente com um mesmo papel semântico sejam adicionadas de uma vez no *corpus* de (re)treinamento. Deveria procurar-se um balanceamento dos papéis semânticos das instâncias selecionadas. Propõe-se modificar a função `selecionar`, de tal forma que invoque uma função chamada de `balancear`, encarregada de realizar esse balanceamento.

Como apresentado no Algoritmo 9, o objetivo da função `balancear` é que o mesmo número de instâncias por papel semântico seja selecionado em cada iteração. Para isso, esta função faz o seguinte:

1. Recebe as instâncias retornadas por `selecionar`, incluindo a etiqueta de papel semântico atribuída (*label*) e a confiança do classificador nessa anotação automática.

2. Separa em D as instâncias que correspondem a cada *label* atribuída.

3. Determina o conjunto D_{label} que possui o menor número de instâncias e toma seu tamanho o como o número de instâncias mínimo a ser considerado na seleção balanceada ($min_{ins-per-label}$).

4. De cada conjunto D_{label}, entrai as $min_{ins-per-label}$ instâncias de maior confiança e as junta para formar o novo conjunto de instâncias selecionadas para o retreinamento.

Na Tabela 6.9 apresentam-se os resultados obtidos pelo sistema semissupervisionado usando esta modificação no processo de seleção.

Algoritmo 9: Função `balancear` do algoritmo *self-training*

Entrada: L_{selec}: instâncias selecionadas, E: etiquetas de papel semântico atribuídas

Saída: $L_{selec-bal}$: instâncias selecionadas balanceadas

para cada $label \in E$ **faça**
  $D_{label} \leftarrow$ instâncias em L_{selec} que possuem a etiqueta $label$
fim
$min_{ins-per-label} \leftarrow$ mínimo de instâncias anotadas para todas as $label$ em E
para cada $label \in D$ **faça**
  $L_{selec-bal} \leftarrow L_{selec-bal}$ + `extrair` $(min_{ins-per-label}, D_{label})$
fim
retorna $L_{selec-bal}$

Tabela 6.9: Resultados globais do sistema semissupervisionado nos dados de teste usando *self-training* com condição de parada simplificada e seleção balanceada.

Tarefa	Precisão	Cobertura	F_1	Acurácia
Identificação	94,4%	94,0%	94,2	–
Classificação	–	–	–	83,0%
Ident. + Class.	80,5%	80.2%	80,4	–

Com relação aos resultados na Tabela 6.7, os apresentados na Tabela 6.9 são levemente melhores. A acurácia na classificação de argumentos é 0,3 unidade melhor, retornando ao valor obtido pelo algoritmo *self-training* básico. Para a tarefa combinada, apresentam-se ganhos nas três medidas de avaliação: 0,7 unidade em precisão, 0,7 unidade em cobertura, e 0,8 unidade em F_1. Estes resultados indicam que a seleção balanceada realmente permite um treinamento mais apropriado do classificador, fornecendo instâncias de retreinamento que ajudam a que possa generalizar melhor e aprimorar as suas anotações automáticas.

Como nos casos anteriores, os resultados do sistema com esta modificação não possuem uma diferença estatisticamente significativa quando comparados com os apresentados na Tabela 6.3 ($p > 0,01$). Contudo, novamente, cumpre-se com aproximá-los aos resultados objetivo da Tabela 6.4.

6.4.3 Seleção Balanceada Auxiliada por Similaridade

No Capítulo 5, foram apresentados modelos não supervisionados que aproveitam as similaridades entre os argumentos de um verbo que possuem um mesmo papel semântico, para agrupá-los apropriadamente em *clusters* que representam algum papel semântico específico. Pretende-se aproveitar esta ideia para melhorar a seleção de instâncias a serem

acrescentadas ao conjunto de retreinamento em cada iteração do algoritmo *self-training*.

Até agora, a confiança na seleção foi dada pela probabilidade do classificador em atribuir um determinado papel semântico a um candidato. Esta confiança poderia ser aumentada ou reduzida, considerando a similaridade que existe entre o candidato sendo avaliado e aqueles que já pertencem ao conjunto de retreinamento.

Basicamente, quando um candidato é avaliado para seleção, calcula-se a similaridade que existe entre este e o *cluster* formado por todas as instâncias (já anotadas) que possuem o mesmo papel semântico atribuído pelo classificador para o verbo alvo da proposição à qual o candidato pertence. Esta similaridade, como no caso não supervisionado, forneceria evidencia positiva ou negativa que aumentaria ou diminuiria, respectivamente, a confiança do classificador.

Para calcular a similaridade, usam-se as fórmulas de agregação por camada e combinação de pontuações de camadas descritas no Capítulo 5. Específicamente, emprega-se a versão modificada de agregação por camada descrita para o modelo de particionamento aglomerativo de grafos multi-camada. Adicionalmente, os limiares mínimos de β e γ, usados nessas fórmulas, foram estabelecidos para 0,75 e não 0 como na versão original. Isto para evitar que baixas similaridades afetem negativamente a confiabilidade do classificador. A função encarregada da atualização de limiares é executada cada vez que não sejam selecionados novos candidatos para o conjunto de retreinamento. A Tabela 6.10 apresenta os resultados obtidos por esta versão do algoritmo *self-training*.

Tabela 6.10: Resultados globais do sistema semissupervisionado nos dados de teste usando *self-training* com condição de parada simplificada e seleção balanceada auxiliada por similaridade.

Tarefa	Precisão	Cobertura	F_1	Acurácia
Identificação	94,4%	94,0%	94,2	–
Classificação	–	–	–	83,2%
Ident. + Class.	80,7%	80,4%	80,5	–

Os resultados obtidos são levemente melhores aos apresentados na Tabela 6.9. Existe um ganho de 0,2 unidade em acurácia para classificação de argumentos, e de 0,1 unidade em F_1 para a tarefa combinada; este último como consequência do incremento em 0,2 unidade da precisão e da cobertura. A diferença entre estes resultados e os da Tabela 6.3 não é estatisticamente signficativa ($p > 0,01$). Porém, estes são mais próximos aos resultados objetivo da Tabela 6.4. O ganho no desempenho do sistema semissupervisionado usando esta versão do *self-training* é mínimo. Contudo, mostrou-se que uma simples incorporação

das medidas de similaridade no método `selecionar` já permite aprimorar um pouco os resultados.

Um possível motivo para que a contribuição da informação de similaridade não tenha resultado em aprimoramentos significativos no desempenho é que a parametrização do algoritmo é complexa. Estimar os limiares mínimos de α, β e γ mais apropriados para *self-training* demandaria realizar *cross-validation* (como em casos anteriormente descritos), o que é temporalmente custoso para esta versão do algoritmo. Contudo, esta abordagem já demonstra ser útil para a anotação semissupervisionada de papeís semânticos.

6.5 Considerações Finais

Neste capítulo apresentou-se uma variedade de experimentos com *self-training*, um algoritmo muito conhecido de aprendizado de máquina semissupervisionado. A simples implementação e entendimento do algoritmo permitiu realizar várias modificações à versão original, as quais demonstraram ser úteis no aprimoramento do desempenho de um classificador supervisionado treinado com poucos dados anotados.

Obteve-se o melhor desempenho nos dados de teste (que correspondem ao *benchmark* descrito no Capítulo 4) com uma versão do algoritmo que inclui: (i) uma condição de parada baseada na confiança mínima da anotação do classificador; (ii) balanceamento no número de instâncias por papel semântico acrescentadas ao *corpus* de retreinamento; e (iii) seleção auxiliada por similaridade entre argumentos de um mesmo verbo.

Os resultados obtidos são melhores (em valor) aos obtidos por um sistema supervisionado treinado com poucos dados anotados, mas a diferença não é estatisticamente significativa. Mais importante, é que o desempenho do sistema semissupervisionado é comparável com o de um sistema supervisionado treinado com um conjunto maior de dados anotados. Estes resultados permitem validar a hipótese planteada nesta dissertação: *é possível empregar técnicas de aprendizado de máquina semissupervisionado para anotar automaticamente com papéis semânticos sentenças escritas em português do Brasil com um desempenho comparável ao de um anotador supervisionado.*

Capítulo

7

Conclusões

Um dos maiores desafios de pesquisa na Anotação de Papéis Semânticos (APS) é desenvolver aplicações para línguas diferentes do inglês. Para o português do Brasil, projetos recentes em semântica lexical fornecem os recursos computacionais necessários para investigação nesta área. Porém, a quantidade de dados anotados disponibilizados não é suficientemente significativa para um aprendizado supervisionado satisfatório. Logo, a hipótese subjacente a esta dissertação considera que é possível empregar uma abordagem semissupervisionada para anotar automaticamente com papéis semânticos sentenças escritas em português do Brasil, atingindo resultados comparáveis aos de um anotador supervisionado treinado para esta língua.

Para comprovar a validade dessa hipótese, implementou-se um anotador automático de papéis semânticos que usa etiquetas do PropBank para o português do Brasil. Empregou-se o algoritmo *self-training* com modelos de Regressão Logística (ou Máxima Entropia) e medidas de similaridade entre os constituintes das sentenças para realizar o aprendizado semissupervisionado. Esta abordagem demonstrou ser capaz de aproveitar a informação fornecida pelos dados anotados e os não anotados com um desempenho estatisticamente comparável ao de um classificador treinado com mais dados anotados.

A seguir, resumem-se as principais contribuições desta dissertação (Seção 7.1), como são os recursos criados e os métodos implementados; e discutem-se possíveis trabalhos futuros (Seção 7.2) que, se forem explorados, beneficiariam grandemente a toda área do Processamento de Língua Natural (PLN) do português do Brasil.

7.1 Contribuições

1. Criou-se um ***benchmark*** para avaliar o desempenho de sistemas de APS para o português do Brasil. Ele está baseado nas CoNLL *Shared Tasks* (STs), oferecendo o mesmo rigor na avaliação e tipos de recursos. Assim, fornecem-se conjuntos de dados de treinamento e teste (derivados do *corpus* PropBank.Br), medidas de avaliação de resultados (calculadas usando o *script* oficial das STs) e um sistema *baseline* baseado em umas poucas regras simples. Empregaram-se estes recursos na implementação dos diferentes sistemas de APS automática apresentados nesta dissertação. Demonstrou-se que este *benchmark* permite comparar objetivamente o desempenho de diferentes abordagens para esta tarefa do PLN. Espera-se que este *benchmark* seja útil na implementação e comparação de diferentes abordagens para APS automática e contribua no avanço do estado da arte da APS para o português.

2. Implementou-se o primeiro **sistema supervisionado** de APS para o português do Brasil. Este sistema de três fases (poda, identificação e classificação de argumentos) emprega o algoritmo de Regressão Logística (ou Máxima Entropia) e um conjunto de atributos rico em informação sintática e lexical dos constituintes das sentenças, para aprender a anotar automaticamente. Usando os recursos fornecidos pelo *benchmark*, demonstrou-se que a abordagem supervisionada – usando os (poucos) dados anotados disponíveis – permite obter resultados próximos aos de sistemas estado-da-arte de outras línguas na tarefa de identificação de argumentos, mas não na classificação de argumentos. Mesmo assim, este sistema constitui-se em uma base sobre a qual modificações podem ser propostas para aprimorar os resultados obtidos pela abordagem supervisionada.

3. Propôs-se uma **abordagem para seleção de atributos** baseada na importância unitária de cada atributo em cada fase do processo de APS automática. Esta abordagem mostrou-se útil para estimar a contribuição individual de cada atributo para cada tarefa (identificação e classificação), assim como para analisar como a interação entre os atributos afeta o desempenho do sistema em cada tarefa. Comprovou-se o já indicado na literatura: (i) os atributos úteis para cada fase da APS são diferentes; (ii) atributos estruturais (como Caminho) são mais úteis na tarefa de identificação de argumentos, enquanto atributos lexicais ou semânticos mais específicos (como Núcleo) são mais importantes na classificação de argumentos. Os atributos selecionados para cada etapa são:

   - **Identificação de Argumentos:** Caminho, Tipo de Sintagma do Irmão Esquerdo e Primeira Palavra + POS da Primeira Palavra.

- **Classificação de Argumentos:** Primeira Palavra + POS da Primeira Palavra, Forma da Primeira Palavra, Lema da Primeira Palavra, Núcleo, Lema do Núcleo, Sequência TOP, Sequência POS, Lema do Predicado + Tipo de Sintagma, Última Palavra + POS da Última Palavra, Lema do Predicado + Caminho, POS da Primeira Palavra, Núcleo do Irmão Esquerdo, Núcleo do Irmão Direito, Voz + Posição, POS do Núcleo do Irmão Esquerdo, Tipo de Sintagma do Irmão Direito, Núcleo do Sintagma Preposicional, Caminho, Saco de Substantivos, Lema da Segunda Palavra, Tipo de Sintagma, Lema do Predicado + Núcleo, POS da Terceira Palavra, Lema do Predicado, POS do Núcleo do Pai, POS da Palavra à Esquerda do Predicado, NEG, POS do Predicado, Número de Sintagmas Verbais, e Número de Orações na Parte Descendente do Caminho.

Os atributos selecionados permitem obter resultados comparáveis aos do sistema que emprega o conjunto completo. Assim, demonstrou-se que uma seleção inteligente dos atributos a serem usados pelo sistema de APS permite reduzir a sua complexidade, sem afetar significativamente o seu desempenho.

4. Criou-se o ***corpus* PropBank.Br com anotação por dependências**. As árvores sintáticas de dependentes foram extraídas do *corpus* PropBank.Br com ajuda das regras[1] elaboradas por Eckhard Bick para a CoNLL-X *Shared Task*; igualmente, as estruturas predicado-argumento (ou papéis semânticos) foram derivadas da anotação por constituintes do PropBank.Br, usando as regras criadas para a CoNLL 2008 *Shared Task*. Empregando a informação fornecida neste *corpus*, demonstrou-se que extrair atributos que refletem a relação de dependência entre o verbo alvo e o núcleo do constituinte candidato a argumento, aprimora significativamente o desempenho do sistema de APS na tarefa de classificação de argumentos.

5. Implementaram-se métodos não supervisionados de **indução de papéis semânticos** e adaptaram-se para o português do Brasil. Usando grafos cujos vértices correspondem aos candidatos a argumentos e cujas arestas expressam a similaridade entre os candidatos, o objetivo dos modelos é particionar os grafos em *clusters* de vértices que representam papéis semânticos específicos para um verbo. Demonstrou-se que os argumentos de um determinado verbo com o mesmo papel semântico possuem similaridades nos níveis sintático e lexical que permitem agrupá-los e diferenciá-los não trivialmente de instâncias que possuem outros papéis semânticos para um verbo em específico. Este comportamento mostrou-se particularmente útil para diferen-

[1] http://ilk.uvt.nl/conll/data/portuguese/README

ciar argumentos de verbos com poucas proposições no *corpus* PropBank.Br com anotação por dependências.

6. Implementou-se um **método semissupervisionado** de APS baseado no algoritmo *self-training* e que usa modelos de Regressão Logística como classificador base. Duas modificações foram realizadas ao algoritmo original no processo de seleção de argumentos anotados automaticamente a serem acrescentados no conjunto de treinamento: balanceamento no número de argumentos por papel semântico e seleção auxiliada por similaridade entre argumentos.

   Demonstrou-se que para realizar um treinamento semissupervisionado apropriado do classificador, é necessário **fornecer instâncias de retreinamento de forma balanceada**, evitando sobrecarregar ao algoritmo com muitas instâncias anotadas com apenas poucos tipos de papeis semânticos. Isto permite ao classificador generalizar melhor o seu aprendizado e aprimorar as suas anotações automáticas.

   O uso da **similaridade entre argumentos de um mesmo verbo** mostrou-se como uma modificação promissora ao algoritmo de *self-training*, porque fornece ao método de seleção de instâncias de re-treinamento de evidência positiva e negativa sobre a anotação. Isto permite anotações automáticas de maior confiabilidade.

   Este método semissupervisionado, por não depender em grande medida dos dados de treinamento, poderia beneficiar a anotação automática de textos em domínios diferentes ao que possui o *corpus* PropBank.Br. Diferentes testes devem ser realizados para validar esta hipótese.

Cumprindo com um dos objetivos específicos estabelecidos inicialmente, todos os recursos e anotadores implementados neste trabalho serão disponibilizados no **PortLex**[2]. Este portal tem a missão de agregar trabalhos relacionados a léxicos computacionais para o português e disponibilizá-los a comunidade científica.

7.2 Trabalhos Futuros

1. Aprimorar o *benchmark* acrescentando informação nos dados fornecidos para avaliar o impacto de usar **árvores sintáticas automáticas** e **dependência de domínio**. No primeiro caso, utilizar-se-ia um *parser* sintático (como o Palavras) para anotar automaticamente as sentenças do *corpus* Bosque e transferir-se-ia apropriadamente a informação de papéis semânticos. No segundo caso, anotar-se-ia um pequeno conjunto de sentenças de um *corpus* de gênero distinto ao *corpus* CETENFolha

[2] http://www2.nilc.icmc.usp.br/portlex/

(gênero jornalístico) e acrescentar-se-ia a mesma informação que os dados originais (atributos morfológicos, árvores sintáticas, etc.).

2. Aprimorar o conjunto de atributos dos constituintes das sentenças com **informação semântica** (como entidades nomeadas). Além disso, extrair **atributos mais específicos ao português** que permitam detetar padrões linguísticos próprios dessa língua. Adicionalmente, usar informação fornecida pela VerbNet.Br para melhorar a anotação dos argumentos de verbos não presentes no *corpus* de treinamento, mas que pertençam à mesma classe na VerbNet.Br de um que esteja presente.

3. Explorar outros **métodos de aprendizado**, como *co-training*, SVMs semissupervisionadas e métodos semissupervisionados baseados em grafos. A representação por grafos mostrou-se útil nos métodos de indução de papéis semânticos, os que obtiveram resultados promissórios, evidenciando que esta abordagem vale a pena ser explorada em melhor profundidade.

4. Acrescentar uma **fase de pós-processamento** para validações pós-anotação; por exemplo, que mais de um constituinte em uma sentença não possua o mesmo papel semântico. Igualmente, implementar um método de **inferência global** de tal forma que a anotação dos candidatos a argumento não seja realizada de forma individual, mas levando em consideração a anotação dos outros constituintes da mesma sentença.

5. Executar uma **avaliação extrínseca** do anotador como parte de um sistema de PLN mais complexo (simplificação, tradução automática, sumarização, etc.).

Existe um crescente interesse na comunidade de PLN no Brasil para desenvolver pesquisas na área de análise semântica. O projeto WordNet.Br tem disponibilizado à comunidade um recurso muito importante e usado no desenvolvimento de várias aplicações. Agora, em conjunto com a VerbNet.Br e o PropBank.Br, espera-se que os recursos criados e os métodos implementados nesta dissertação contribuam par aumentar o interesse no desenvolvimento de aplicações para análise semântica e beneficie a muitas outras áreas do Processamento de Língua Natural do português do Brasil.

Referências Bibliográficas

Abend, O. e Rappoport, A. (2010). Fully Unsupervised Core-Adjunct Argument Classification. In *48th Annual Meeting of the ACL*, páginas 226–236, Uppsala, Sweden. ACL.

Abend, O., Reichart, R., e Rappoport, A. (2009). Unsupervised argument identification for Semantic Role Labeling. In *47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, páginas 28–36, Singapore.

Abney, S. (2007). *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC, 1ª edição.

Aluísio, S. M., Pinheiro, G. M., Manfrim, A. M. P., Genovês Jr., L. H. M., e Tangin, S. E. O. (2004). The Lacio-web: Corpora and Tools to Advance Brazilian Portuguese Language Investigations and Computational Linguistic Tools. In *4th International Conference on Language Resources and Evaluation*, páginas 1779–1782, Lisbon, Portugal.

Alva-Manchego, F. e Rosa, J. L. (2012a). Semantic Role Labeling for Brazilian Portuguese: A Benchmark. In Pavón, J., Duque-Méndez, N., e Fuentes-Fernández, R., editors, *IBERAMIA 2012*, volume 7637 of *LNAI*, páginas 481–490. Springer, Heidelberg.

Alva-Manchego, F. e Rosa, J. L. (2012b). Towards Semi-supervised Brazilian Portuguese Semantic Role Labeling: Building a Benchmark. In Caseli, H., Villavicencio, A., Teixeira, A., e Perdigão, F., editors, *PROPOR 2012*, volume 7243 of *LNAI*, páginas 210–217. Springer, Heidelberg.

Amancio, M. A., Duran, M. S., e Aluisio, S. M. (2010). Automatic question categorization: a new approach for text elaboration. In *Workshop in Natural Language Processing and*

web-based Technologies 2010, in conjunction with IBERAMIA 2010, páginas 21–30, Bahía Blanca, Argentina.

Aziz, W. e Specia, L. (2011). Fully automatic compilation of portuguese-english and portuguese-spanish parallel corpora. In *8th Brazilian Symposium in Information and Human Language Technology*, Cuibá, MT, Brazil.

Baker, C. F., Fillmore, C. J., e Lowe, J. B. (1998). The Berkeley FrameNet Project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, volume 1, páginas 86–90, Montreal, Quebec, Canada. ACL.

Bick, E. (2000). *The parsing system PALAVRAS: automatic grammatical analysis of Portuguese in a constraint grammar framework*. Aarhus University Press.

Bick, E. (2007). Automatic Semantic Role Annotation for Portuguese. In *5th Workshop on Information and Human Language Technology*, páginas 1713–1716, Rio de Janeiro, Brazil.

Blum, A. e Mitchell, T. (1998). Combining Labeled and Unlabeld Data with Co-Training. In *11th Annual Conference on Computational Learning Theory*, páginas 92–100, Madison, WI.

Branco, A., Carvalheiro, C., Pereira, S., Silveira, S., Silva, J., Castro, S., e Graça, J. (2012). A propbank for portuguese: the cintil-propbank. In *Eight International Conference on Language Resources and Evaluation*, páginas 1516–1521, Istanbul, Turkey.

Branco, A. e Costa, F. (2010). A deep linguistic processing grammar for portuguese. In *Lecture Notes in Artificial Intelligence*, volume 6001 of *86–89*. Springer, Berlin.

Branco, A., Costa, F., Silva, J., Silveira, S., Castro, S., Avelãs, M., Pinto, C., e Graça, J. (2010). Developing a deep linguistic databank supporting a collection of treebanks: the cintil deepgrambank. In *7th International Conference on Language Resources and Evaluation*, páginas 1810–1815, Valletta, Malta.

Buchholz, S. e Marsi, E. (2006). CoNLL-X Shared Task on Multilingual Dependency Parsing. In *10th Conference on Computational Natural Language Learning*, páginas 149–164, New York City. ACL.

Burchardt, A., Erk, K., Frank, A., Kowalski, A., e Padó, S. (2006). SALTO - A Versatile Multi-Level Annotation Tool. In *Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, páginas 517–520, Genoa, Italy.

Carlson, A., Cumby, C., Rosen, J., e Roth, D. (1999). The SNoW Learning Architecture. Relatório Técnico UIUCDCS-R-99-2101, University of Illinois, Urbana/Champaign, Urbana, Illinois.

Carreras, X. e Màrquez, L. (2004). Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In *8th Conference on Computational Natural Language Learning: Shared Task*, páginas 89–97, Boston, MA, USA. ACL.

Carreras, X. e Màrquez, L. (2005). Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *9th Conference on Computational Natural Language Learning: Shared Task*, páginas 152–164, Ann Arbor, Michigan. ACL.

Caseli, H., Pereira, T., Specia, L., Pardo, T., Gasperin, C., e Aluísio, S. (2009). Building a Brazilian Portuguese Parallel Corpus of Original and Simplified Texts. In Gelbukh, A., editor, *10th Conference on Intelligent Text Processing and Computational Linguistics*, volume 41 of *Advances in Computational Linguistics, Research in Computer Science*, páginas 59–70.

Charniak, E. (2000). A Maximum-Entropy-Inspired Parser. In *1st Conference of the North American Chapter of the Association for Computational Linguistics*, páginas 132–139, Seattle, Washington.

Charniak, E. e Johnson, M. (2005). Coarse-to-fine n-best Parsing and Maxent Discriminative Reranking. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, páginas 173–180, Ann Arbor, MI.

Che, W., Li, Z., Li, Y., Guo, Y., Qin, B., e Liu, T. (2009). Multilingual dependency-based syntactic and semantic parsing. In *13th Conference on Computational Natural Language Learning: Shared Task*, páginas 49–54, Boulder, Colorado. ACL.

Clark, S., Curran, J. R., e Osborne, M. (2003). Bootstrapping POS Taggers Using Unlabelled Data. In *7th Conference on Natural Language Learning (CoNLL'03) at HLT-NAACL 2003*, volume 4, páginas 49–55, Edmonton, Canada. ACL.

Collins, M. (1999). *Head-driven Statistical Models for Natural Language Parsing*. Ph.d. thesis, University of Pennsylvania, Philadelphia.

Collins, M. e Koo, T. (2005). Discriminative Reranking for Natural Language Parsing. *Computational Linguistics*, 31(1):25–69.

Cook, W. A. (1989). *Case Grammar Theory*. Georgetown University Press.

Diab, M., Moschitti, A., e Pighin, D. (2008). Semantic Role Labeling Systems for Arabic using Kernel Methods. In *Proceedings of ACL-08: HLT*, páginas 798–806, Columbus, Ohio. ACL.

Dias-da-Silva, B. (1996). *A face tecnológica dos estudos da linguagem: o processamento automático das línguas naturais*. Tese de doutorado, Faculdade de Ciências e Letras, Universidade Estadual Paulista - UNESP, Araraquara.

Dias-da-Silva, B. (2004). Wordnet.br: an exercise of human language technology research. *PaLavra*, 12:15–24.

Dias-da-Silva, B., Di Felippo, A., e Hasegawa, R. (2006). Methods and Tools for Encoding the WordNet.Br Sentences, Concept Glosses, and Conceptual-Semantic Relations. In Vieira, R., Quaresma, P., Nunes, M., Mamede, N., Oliveira, C., e Dias, M., editors, *Computational Processing of the Portuguese Language*, volume 3960 of *LNCS*, páginas 120–130. Springer Berlin / Heidelberg.

Dias-da-Silva, B. C., Oliveira, M. F. d., e Moraes, H. R. d. (2002). Groundwork for the Development of the Brazilian Portuguese Wordnet. In RANCHHOD, E. and MAMEDE, N. J., editor, *Third International Conference on Advances in Natural Language Processing*, páginas 189–196, London, UK. Springer-Verlag.

Dowty, D. (1991). Thematic proto-roles and argument selection. *Language*, 67(3):547–619.

Duran, M. S. e Aluísio, S. M. (2012). Propbank-Br: a Brazilian treebank annotated with semantic role labels. In *8th International Conference on Language Resources and Evaluation (LREC 2012)*, páginas 1862–1867, Istanbul, Turkey.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Fillmore, C. (1982). Frame Semantics. In *Linguistics in the Morning Calm*, páginas 111–138, Hanshin, Seoul. Linguistics Society of Korea.

Fillmore, C. (1985). Frames and the Semantics of Understanding. *Quaderni di Semantica*, 6(2):222–254.

Fillmore, C., Bach, E., e Harms, R. (1968). *The Case for Case*. Holt, Rinhehart and Winston.

Fillmore, C. J. (1976). Frame Semantics and the Nature of Language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, páginas 20–32.

Fillmore, C. J., Johnson, C. R., e Petruck, M. R. (2003). Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250.

Fleischman, M., Kwon, N., e Hovy, E. (2003). Maximum entropy models for FrameNet classification. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, páginas 49–56, Morristown, NJ, USA. ACL.

Fonseca, E. R. e Rosa, J. L. (2012). An Architecture for Semantic Role Labeling on Portuguese. In Caseli, H., Villavicencio, A., Teixeira, A., e Perdigão, F., editors, *PROPOR 2012*, volume 7243 of *LNAI*, páginas 204–209. Springer, Heidelberg.

Frank, A., Krieger, H., Xu, F., Uszkoreit, H., Crysmann, B., Jórg, B., e Scháfer, U. (2007). Question answering from structured knowledge sources. *Journal of Applied Logic*, 5(1):20–48.

Fürstenau, H. e Lapata, M. (2009a). Graph Alignment for Semi-Supervised Semantic Role Labeling. In *2009 Conference on Empirical Methods in Natural Language Processing*, páginas 11–20, Singapore. ACL and AFNLP.

Fürstenau, H. e Lapata, M. (2009b). Semi-supervised Semantic Role Labeling. In *12th Conference of the European Chapter of the ACL*, páginas 220–228, Athens. ACL.

Fürstenau, H. e Lapata, M. (2012). Semi-supervised Semantic Role Labeling via Structural Alignment. *Computational Linguistics*, 38(1):135–171.

Gildea, D. e Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Giménez, J. e Màrquez, L. (2007). Linguistic features for automatic evaluation of heterogenous MT systems. In *Second Workshop on Statistical Machine Translation*, páginas 256–264. ACL.

Giménez, J. e Màrquez, L. (2008). A smorgasbord of features for automatic MT evaluation. In *Third Workshop on Statistical Machine Translation*, páginas 195–198. ACL.

Hacioglu, K., Pradhan, S., Ward, W., Martin, J. H., e Jurafsky, D. (2004). Semantic Role Labeling by Tagging Syntactic Chunks. In *Proceedings of Conference on Computational Natural Language Learning (CoNLL) 2004*, páginas 110–113.

Hajic, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Mayers, A., Nivre, J., Padó, S., Stepánek, J., Stranák, P., Surdeanu, M., Xue, N.,

e Zhang, Y. (2009). The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *13th Conference on Computational Natural Language Learning: Shared Task*, páginas 1–18, Boulder, CO, USA. ACL.

He, S. e Gildea, D. (2004). Semantic Labeling by Maximum Entropy Model. Relatório técnico, The University of Rochester, Rochester, New York.

He, S. e Gildea, D. (2007). Self-training and Co-training for Semantic Role Labeling: Primary Report. Relatório Técnico 891, The University of Rochester.

Hofmann, T. e Puzicha, J. (1998). Statistical models for co-occurrence data. Relatório técnico, Massachusetts Institute of Technology, Cambridge, MA, USA.

Johansson, R. e Nugues, P. (2006). A FrameNet-based semantic role labeler for Swedish. In *COLING/ACL 2006 Main conference poster sessions*, páginas 436–443, Stroudsburg, PA, USA.

Johansson, R. e Nugues, P. (2008). Dependency-based Syntactic-Semantic Analysis with PropBank and NomBank. In *12th Conference on Computational Natural Language Learning - Shared Task*, páginas 183–187, Manchester, United Kingdom. ACL.

Kipper, K., Korhonen, A., Ryant, N., e Palmer, M. (2006). Extending VerbNet with Novel Verb Classes. In *5th international conference on Language Resources and Evaluation (LREC 2006)*, páginas 1027–1032, Genova, Italy.

Kipper-Schuler, K. (2005). *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. Tese de Doutorado, University of Pennsylvania.

Korhonen, A. e Briscoe, T. (2004). Extended Lexical-Semantic Classification of English Verbs. In *HLT/NAACL Workshop on Computational Lexical Semantics*, páginas 38–45, Boston, MA.

Lang, J. (2012). *Unsupervised Induction of Semantic Roles*. Tese de Doutorado, School of Informatics, University of Edinburgh.

Lang, J. e Lapata, M. (2010). Unsupervised Induction of Semantic Roles. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, páginas 939–947, Los Angeles, California. ACL.

Lang, J. e Lapata, M. (2011a). Unsupervised Semantic Role Induction via Split-Merge Clustering. In *49th Annual Meeting of the Association for Computational Linguistics*, páginas 1117–1126, Portland, Oregon. ACL.

Lang, J. e Lapata, M. (2011b). Unsupervised Semantic Role Induction with Graph Partitioning. In *2011 Conference on Empirical Methods in Natural Language Processing*, páginas 1320–1331, Edinburgh, Scotland, UK. ACL.

Lee, J.-Y., Song, Y.-I., e Rin, H.-C. (2007). Investigation of Weakly Supervised Learning for Semantic Role Labeling. In *Sixth International Conference on Advanced Language Processing and Web Information Technology*, páginas 165–170, Luoyang, Henan, China.

Levin, B. (1993). English verb classes and alternations: A preliminary investigation. *Chicago, Il*.

Lima, M. C. P. B. (1982). A Gramática dos Casos e o "Dativo". *Alfa*, 26:33–46.

Lin, D. (1998). Automatic Retrieval and Clustering of Similar Words. In *17th International Conference on Computational Linguistics and 36th Annual Meeting of the ACL (COLING/ACL)*, páginas 768–774, Montreal, Canada.

Litkowski, K. (2004). Senseval-3 task: Automatic Labeling of Semantic Roles. In Mihalcea, R. e Edmonds, P., editors, *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, páginas 9–12, Barcelona, Spain. ACL.

Loper, E., Yi, S., e Palmer, M. (2007). Combining lexical resources: Mapping between PropBank and VerbNet. In *7th International Workshop on Computational Linguistics*, páginas 1–12, Tilburg, The Netherlands.

Manning, C. D., Raghavan, P., e Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Marcus, M. P., Santorini, B., e Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Màrquez, L. (2009). Semantic role labeling: past, present and future. In *Tutorial Abstracts of ACL-IJCNLP 2009: 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, páginas 3–3, Morristown, NJ, USA. ACL.

Màrquez, L., Carreras, X., Litkowski, K. C., e Stevenson, S. (2008). Semantic Role Labeling: An Introduction to the Special Issue. *Computational Linguistics*, 34(2):145–159.

Màrquez, L., Villarejo, L., Martí, M. A., e Taulé, M. (2007). Semeval-2007 task 09: Multilevel semantic annotation of catalan and spanish. In *4th International Workshop on Semantic Evaluations*, páginas 42–47, Morristown, NJ, USA. ACL.

Martí, M. A. e Taulé, M. (2007). CESS-ECE. Corpus Anotados del Español y Catalán. *Arena Romanística*, (1). Monografía dedicada a Corpus and text linguistics in Romance languages.

McClelland, J. L. e Kawamoto, A. H. (1986). *Mechanisms of sentence processing: assigning roles to constituents*, páginas 272–325. MIT Press, Cambridge, MA, USA.

Melli, G., Wang, Y., Liu, Y., Kashani, M. M., Shi, Z., Gu, B., Sarkar, A., e Popowich, F. (2005). Description of Squash, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task. In *2005 Document Understanding Conference*, Vancouver, B.C., Canada.

Minsky, M. (1975). A Framework for Representing Knowledge. In Winston, P. H., editor, *The Psychology of Computer Vision*. McGraw-Hill, NY, NY.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Monachesi, P., Stevens, G., e Trapman, J. (2007). Adding semantic role annotation to a corpus of written Dutch. In *1st Linguistic Annotation Workshop*, páginas 77–84, Prague, Czech Republic. ACL.

Monard, M. C. e Baranauskas, J. A. (2003). *Sistemas Inteligentes - Fundamentos e Aplicações*, chapter Conceitos sobre aprendizado de máquina. Manole.

Morante, R. e Bosch, A. V. D. (2009). Feature Construction for Memory-Based Semantic Role Labeling of Catalan and Spanish. In Nicolov, N., Angelova, G., e Mitkov, R., editors, *Recent Advances in Natural Language Processing V*, volume 309, páginas 131–142, Amsterdam.

Morante, R. e Busser, B. (2007). ILK2: semantic role labelling for Catalan and Spanish using TiMBL. In *4th International Workshop on Semantic Evaluations*, páginas 183–186, Stroudsburg, PA, USA. ACL.

Moreda, P., Navarro, B., e Palomar, M. (2007). Corpus-based semantic role approach in information retrieval. *Data & Knowledge Engineering*, 61(3):467–483.

Moreda Pozo, P. (2008). *Los Roles Semánticos en la Tecnología del Lenguaje Humano: Anotación y Aplicación*. Doctoral thesis, Universidad de Alicante.

Muniz, M., Paulovich, F. V., Minghim, R., Infante, K., Muniz, F., Vieira, R., e Aluísio, S. (2007). Taming the tiger topic: An xces compliant corpus portal to generate subcorpora based on automatic text-topic identification. In *Corpus Linguistic Conference*, Birmingham.

Padó, S. (2006). *User's guide to `sigf`: Significance testing by approximate randomisation.*

Palmer, M., Gildea, D., e Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Palmer, M., Gildea, D., e Xue, N. (2010). *Semantic Role Labeling*, volume 3. Morgan & Claypool Publishers.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., e Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python . *Journal of Machine Learning Research*, 12:2825–2830.

Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J. H., e Jurafsky, D. (2005). Support vector learning for semantic argument classification. *Machine Learning*, 60(1):11–39.

Pradhan, S. S., Ward, W., e Martin, J. H. (2008). Towards Robust Semantic Role Labeling. *Computational Linguistics*, 34(2):289–310.

Punyakanok, V., Koomen, P., Roth, D., e Yih, W.-t. (2005). Generalized inference with multiple semantic role labeling systems. In *9th Conference on Computational Natural Language Learning*, páginas 181–184, Stroudsburg, PA, USA. ACL.

Punyakanok, V., Roth, D., e tau Yih, W. (2008). The Importance of Syntactic Parsing and Inference in Semantic Role Labeling. *Computational Linguistics*, 34(2):257–287.

Rosa, J. L. G. (2007). A connectionist thematic grid predictor for pre-parsed natural language sentences. In Liu, D., Fei, S., Hou, Z., Zhang, H., e Sun, C., editors, *Advances in Neural Networks ISNN 2007*, volume 4492 of *Lecture Notes in Computer Science*, páginas 825–834. Springer Berlin / Heidelberg.

Rosa, J. L. G. (2008). Aplicação de Técnicas de Aprendizado de Máquina e Linguística Computacional para Tratamento de Textos. Projeto FAPESP nro. 2008/08245-4.

Rosa, J. L. G. e Adán-Coello, J. M. (2010). Biologically plausible connectionist prediction of natural language thematic relations. *Journal of Universal Computer Science*, 16(21):3245–3277.

Salomão, M. M. M. (2009). FrameNet Brasil: um trabalho em progresso. *Calidoscópio*, 7(3):171–182.

Santos, D., Bick, E., e Afonso, S. (2007). Floresta sintá(c)tica: apresentação e história do projecto. *Encontro Um passeio pela Floresta Sintá(c)tica*.

Scarton, C. e Aluísio, S. (2012). Towards a cross-linguistic VerbNet-style lexicon for Brazilian Portuguese. In *LREC 2012 Workshop on Creating Cross-language Resources for Disconnected Languages and Styles*, páginas 11–18, Istanbul, Turkey.

Sequeira, J., Gonçalves, T., e Quaresma, P. (2012). Semantic Role Labeling for Portuguese – A Preliminary Approach –. In Caseli, H., Villavicencio, A., Teixeira, A., e Perdigão, F., editors, *PROPOR 2012*, volume 7243 of *LNAI*, páginas 193–203. Springer, Heidelberg.

Shamsfard, M. e Mousavi, M. S. (2008). Thematic Role Extraction Using Shallow Parsing. *International Journal of Information and Mathematical Sciences*, 4(2):126–132.

Shen, D. e Lapata, M. (2007). Using Semantic Roles to Improve Question Answering. In *EMNLP-CoNLL 2007*, páginas 12–21, Prague, Czech Republic. ACL.

Stenchikova, S., Hakkani-Tür, D., e Tur, G. (2006). QASR: Spoken Question Answering Using Semantic Role Labeling. In *International Conference on Spoken Language Processing (ICSLP)*, páginas 1185–1188, Pittsburgh, Pennsylvania.

Stoyanchev, S., Song, Y., e Lahti, W. (2008). Exact phrases in information retrieval for question answering. In *Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering*, páginas 9–16. ACL.

Strube de Lima, V. L., Nunes, M., e Vieira, R. (2007). Desafios do Processamento de Línguas Naturais. In *SEMISH - XXXIV Seminário Integrado de Software e Hardware. Anais do XXVII Congresso da SBC*, páginas 2202–2216.

Suanmali, L., Binwahlan, M., e Salim, N. (2010). SRL-GSM: A Hybrid Approach based on Semantic Role Labeling and General Statistic Method for Text Summarization. *Journal of Applied Sciences*, 10(3):166–173.

Surdeanu, M., Harabagiu, S., Williams, J., e Aarseth, P. (2003). Using predicate-argument structures for information extraction. In *41st Annual Meeting of the ACL*, volume 1, páginas 8–15, Stroudsburg, PA, USA. ACL.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., e Nivre, J. (2008a). The CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In *12th*

Conference on Computational Natural Language Learning, páginas 159–177, Manchester, UK. ACL.

Surdeanu, M., Morante, R., e Màrquez, L. (2008b). Analysis of Joint Inferences Strategies for the Semantic Role Labeling of Spanish and Catalan. In Gelbukh, A., editor, *CICLing 2008*, volume 4919 of *LNCS*, páginas 206–218. Springer, Heidelberg.

Surdenau, M., Màrquez, L., Carreras, X., e Comas, P. R. (2007). Combination strategies for semantic role labeling. *Journal of Artificial Intelligence Research*, (29):105–151.

Toutanova, K., Haghighi, A., e Manning, C. D. (2008). A Global Joint Model for Semantic Role Labeling. *Computational Linguistics*, 34(2):161–191.

VISIL (2012). Grammatical categories (tags) used in the Floresta project. http://beta.visl.sdu.dk/visl/pt/info/symbolset-floresta.html. Última visita: Julho do 2012.

Waltz, D. e Pollack, J. (1985). Massively parallel parsing: A strongly interactive model of natural language interpretation*. *Cognitive Science*, 9(1):51–74.

Widdows, D. e Cohen, T. (2010). The Semantic Vector Package: New Algorithms and Public Tools for Distributional Semantics. In *Fourth IEEE International Conference on Semantic Computing*, páginas 9–15, Pittsburgh, Pennsylvania.

Wu, D. e Fung, P. (2009a). Can Semantic Role Labeling Improve SMT. In *13th Annual Conference of the European Association for Machine Translation*, páginas 218–225, Barcelona, May.

Wu, D. e Fung, P. (2009b). Semantic roles for SMT: A hybrid two-pass model. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, páginas 13–16, Boulder, CO, USA. ACL.

Xue, N. (2008). Labeling Chinese Predicates with Semantic Roles. *Computational Linguistics*, 34(2):225–255.

Xue, N. e Palmer, M. (2004). Calibrating Features for Semantic Role Labeling. In *2004 Conference on Empirical Methods in Natural Language Processing*, páginas 88–94, Barcelona, Spain. ACL.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting on Association for Computational Linguistics*, páginas 189–196, Morristown, NJ, USA. ACL.

Zadeh Kaljahi, R. S. (2010). Adapting self-training for semantic role labeling. In *ACL 2010 Student Research Workshop*, páginas 91–96, Morristown, NJ, USA. ACL.

Zanette, A., Scarton, C., e Zilio, L. (2012). Automatica extraction of subcategorization frames from corpora: an approach to portuguese. In *Demostration Sesion of the Intenational Conference on Computational Processing of Portuguese Language*, Coimbra, Portugal.

Zhao, H., Chen, W., Kit, C., e Zhou, G. (2009). Multilingual dependency learning: a huge feature engineering method to semantic dependency parsing. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, páginas 55–60, Morristown, NJ, USA. ACL.

Zhu, X. e Goldberg, A. (2009). Introduction to semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–130.

Apêndice

A

Equivalência entre Abreviaturas e Nomes de Atributos

Abreviatura usada pelo sistema (inglês)	Nome real do atributo (português)
first_form+first_postag	Forma da Primeira Palavra + POS da Primeira Palavra
first_form	Forma da Primeira Palavra
first_lemma	Lema da Primeira Palavra
head	Núcleo
head_lemma	Lema do Núcleo
top_sequence	Sequência TOP
postag_sequence	Sequência POS
last_form+last_postag	Forma da Última Palavra + POS da Última Palavra
pred_lemma+phrase_type	Lema do Predicado + Tipo de Sintagma
pred_lemma+path	Lema do Predicado + Caminho
first_postag	POS da Primeira Palavra
left_head	Núcleo do Irmão Esquerdo
right_head	Núcleo do Irmão Direito
head_postag	POS do Núcleo
voice+position	Voz + Posição
left_head_postag	POS do Núcleo do Irmão Esquerdo
left_phrase	Tipo de Sintagma do Irmão Esquerdo
right_phrase	Tipo de Sintagma do Irmão Direito

second_form	Forma de Segunda Palavra
preposition	Núcleo do Sintagma Preposicional
bag_of_nouns	Saco de Substantivos
right_head_postag	POS do Núcleo do Irmão Direito
position	Posição
second_lemma	Lema da Segunda Palavra
path	Caminho
phrase_type	Tipo de Sintagma
bag_of_adv	Saco de Advérbios
pred_lema+head	Lema do Predicado + Núcleo
third_form	Forma da Terceira Palavra
third_lemma	Lema da Terceira Palavra
second_postag	POS da Segunda Palavra
punct_left	Pontuação à Esquerda
third_postag	POS da Terceira Palavra
pred_lemma	Lema do Predicado
partial_path	Caminho Parcial
num_clauses_asc	Número de Orações na Parte Ascendente do Caminho
bag_of_adj	Saco de Adjetivos
parent_phrase	Tipo de Sintagma do Pai
punct_right	Pontuação à Direita
parent_head_postag	POS do Núcleo do Pai
pred_context_left_postag	POS da Palavra à Esquerda do Predicado
pred_context_right_postag	POS da Palavra à Direita do Predicado
negation	NEG
num_clauses	Número de Orações
se_in_vp	SE na Oração do Verbo
num_vp_asc	Número de Sintagmas Verbais na Parte Ascendente do Caminho
pred_postag	POS do Predicado
num_vp	Número de Sintagmas Verbais
tree_distance	Distância em Constituintes na Árvore
num_vp_desc	Número de Sintagmas Verbais na Parte Descendente do Caminho
voice	Voz
num_clauses_desc	Número de Orações na Parte Descendente do Caminho
pred_context_right	Palavra à Direita do Predicado
pred_context_left	Palavra à Esquerda do Predicado
subcat	Subcategorização
parent_head	Núcleo do Pai
pred_form	Forma do Predicado

Apêndice

B

Regras de Identificação de Argumentos para Indução de Papéis Semânticos

Aqui são especificados os conjuntos completos de relações usados pelas regras de identificação de argumentos dadas para o português do Brasil na Tabela 5.4. Os símbolos ↑ e ↓ indicam a direção da relação de dependência (para cima e para baixo, respectivamente). As etiquetas das relações sintáticas são as empregadas na anotação manual do corpus Bosque da Floresta Sintá(c)tica. Uma explicação detalhada de cada etiqueta, assim com exemplos de uso de cada uma, pode ser encontrada em VISIL (2012).

As relações na Regra 2 são CO↑↓, PU↑↓, ACC↑, DAT↑, PIV↑, P<↑, ADVL↑, ADVO↑, SUB↑↓, SUB↓ , STA↑, QUE↑, COM↑, EXC↑, SUBJ↑.

As relações na Regra 4 são ADVL↑↓, ADVO↑↓, >A↑↓, A<↑↓, APP↑↓, CJT↑↓, PCJT↑↓, CO↑↓, PASS↑↓, >N↑↓, N<↑↓, ACC↑↓, DAT↑↓, PIV↑↓, PRED↑↓, SUBJ↑↓, SUB↑↓, VOC↑↓.