Pontifícia Universidade Católica
do Rio de Janeiro

Arthur Beltrão Castilho Neto

Anotador de Papeis Semânticos para Português

Dissertação de Mestrado

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

Orientador: Prof. Ruy Luiz Milidiú

Rio de Janeiro,

Dezembro de 2016.

Pontifícia Universidade Católica
do Rio de Janeiro

Arthur Beltrão Castilho Neto

Anotador de Papeis Semânticos para Português

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

Prof. Ruy Luiz Milidiú
Orientador
Departamento de Informática - PUC-Rio

Prof. Marco Antônio Casanova
Departamento de Informática - PUC-Rio

Prof.ª Maria Cláudia de Freitas
Departamento de Letras - PUC-Rio

Prof. Márcio da Silveira Carvalho
Coordenador Setorial do Centro Técnico Científico - PUC-Rio

Rio de Janeiro, 16 de dezembro de 2016.

Arthur Beltrão Castilho Neto

Graduou-se em Tecnólogo em Processamento de Dados na PUC-Rio em 2000, Pós-graduado em Análise, Projeto e Gerência de Sistemas pela PUC-Rio (CCE) em 2003. É Analista de Sistemas do Instituto Brasileiro de Geografia e Estatística (IBGE) onde é responsável pelo desenvolvimento de aplicações de coleta e controle para diversas pesquisas realizadas pelo órgão. Seus interesses incluem engenharia de software, inteligência artificial e processamento de linguagens naturais.

Ficha Catalográfica

Castilho Neto, Arthur Beltrão

Anotador de papeis semânticos para português / Arthur Beltrão Castilho Neto ; orientador: Ruy Luiz Milidiú. – 2016.
78 f.: il. color ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2016.
Inclui bibliografia

1. Informática – Teses. 2. Aprendizado de máquina supervisionado. 3. Processamento de linguagem natural. 4. Anotação de papéis semânticos em português. 5. Modelo linear SVM. 6. Regularização de domínio. I. Milidiú, Ruy Luiz. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Ao meu orientador, o Prof. Ruy Luiz Milidiú, por me prover ensinamentos e oportunidades importantes na minha vida acadêmica e profissional, sem os quais não seria possível concluir esse trabalho.

À minha esposa Cintia pelo amor, carinho e companheirismo nos momentos mais difíceis.

À minha filha Thuany que mesmo nas minhas piores fases sempre soube me levantar com um simples sorriso e um abraço.

Aos meus pais, Artur e Tânia, que me deram o dom da vida e me ensinaram as virtudes fundamentais.

À minha tia Rosane por sua ajuda inestimável durante minha toda minha vida acadêmica na PUC-Rio.

À Prof.ª. Maria Cláudia de Freitas por sua ajuda conhecimento linguístico.

Ao IBGE pela oportunidade oferecida, em especial pela concessão da licença e confiança depositada em mim.

À PUC-Rio e a CAPES pelo apoio financeiro.

Aos meus colegas do laboratório LEARN pelo apoio e suporte na retirada de dúvidas.

Resumo

Castilho Neto, Arthur Beltrão; Milidiú, Ruy. **Anotador de Papeis Semânticos para Português.** Rio de Janeiro, 2016. 78p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A anotação de papeis semânticos (APS) é uma importante tarefa do processamento de linguagem natural (PLN), que possibilita estabelecer uma relação de significado entre os eventos descritos em uma sentença e seus participantes. Dessa forma, tem o potencial de melhorar o desempenho de inúmeros outros sistemas, tais como: tradução automática, correção ortográfica, extração e recuperação de informações e sistemas de perguntas e respostas, uma vez que reduz as ambiguidades existentes no texto de entrada. A grande maioria dos sistemas de APS publicados no mundo realiza a tarefa empregando técnicas de aprendizado supervisionado e, para obter melhores resultados, usam corpora manualmente revisados de tamanho considerável. No caso do Brasil, o recurso lexical que possui anotações semânticas (Propbank.br) é muito menor. Por isso, nos últimos anos, foram feitas tentativas de melhorar esse resultado utilizando técnicas de aprendizado semisupervisionado ou não-supervisionado. Embora esses trabalhos tenham contribuido direta e indiretamente para a área de PLN, não foram capazes de superar o desempenho dos sistemas puramente supervisionados. Este trabalho apresenta uma abordagem ao problema de anotação de papéis semânticos no idioma português. Utilizamos aprendizado supervisionado sobre um conjunto de 114 atributos categóricos e empregando duas técnicas de regularização de domínio, combinadas para reduzir o número de atributos binários em 96%. O modelo gerado usa uma *support vector machine* com solver *L2-loss dual support vector classification* e é testado na base PropBank.br, apresentando desempenho ligeiramente superior ao estado-da-arte. O sistema é avaliado empiricamente pelo script oficial da *CoNLL 2005 Shared Task,* obtendo 82,17% de precisão, 82,88% de cobertura e 82,52% de F1 ao passo que o estado-da-arte anterior atinge 83,0% de precisão, 81,7% de cobertura e 82,3% de F1.

Palavras-chave

Anotação de papéis semânticos; APS; Aprendizado supervisionado; Processamento de língua natural; PLN; Support Vector Machine; SVM; Liblinear; Propbank.br; Regularização de domínio; Seleção de atributos;

Abstract

Castilho Neto, Arthur Beltrão; Milidiú, Ruy (Advisor). **Semantic Role-Labeling for Portuguese.** Rio de Janeiro, 2016. 78p. MSc. Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Semantic role-labeling (SRL) is an important task of natural language processing (NLP) which allows establishing meaningful relationships between events described in a given sentence and its participants. Therefore, it can potentially improve performance on a large number of NLP systems such as automatic translation, spell correction, information extraction and retrieval and question answering, as it decreases ambiguity in the input text. The vast majority of SRL systems reported so far employed supervised learning techniques to perform the task. For better results, large sized manually reviewed corpora are used. The Brazilian semantic role labeled lexical resource (Propbank.br) is much smaller. Hence, in recent years, attempts have been made to improve performance using semi supervised and unsupervised learning. Even making several direct and indirect contributions to NLP, those studies were not able to outperform exclusively supervised systems. This paper presents an approach to the SRL task in Portuguese language using supervised learning over a set of 114 categorical features. Over those, we apply a combination of two domain regularization methods to cut binary features down to 96%. We test a SVM model (L2-loss dual support vector classification) on PropBank.Br dataset achieving results slightly better than state-of-the-art. We empirically evaluate the system using official CoNLL 2005 Shared Task script pulling 82.17% precision, 82.88% coverage and 82.52% F1. The previous state-of-the-art Portuguese SRL system scores 83.0% precision, 81.7% coverage and 82.3% F1.

Keywords

Semantic Role-Labeling; SRL; Supervised learning; Natural Language Processing; NLP; Support Vector Machine; SVM; Liblinear; Propbank.br; Domain regularization; Feature selection;

Sumário

Lista de figuras

Lista de tabelas

Lista de algoritmos

1. Introdução

Desde a criação do primeiro computador digital eletrônico, ENIAC, em 1946 o ser humano vem buscando formas cada vez mais eficazes e fáceis de interagir com as máquinas. A evolução dessas interações entre homens e máquinas é notória, iniciando pelas interfaces de hardware apenas para engenheiros em 1950, seguindo seu curso com as interfaces de programação em COBOL e FORTRAN em 1960, linha de comando em 1970, interação visual e multimídia em 1980, e culminando nas telas sensíveis ao toque e integração total de sensores e periféricos que temos hoje em dia ao alcance das mãos.

Dentre tantas formas disponíveis que temos para realizar a interface com os computadores uma que desperta especial interesse é o uso das ditas linguagens naturais, ou, colocado de forma mais simples, as línguas que usamos em nosso dia a dia, como português inglês e espanhol.

O processamento de linguagens naturais (PLN) é uma área multidisciplinar que relaciona inteligência artificial, linguística computacional e interação humano-computador (IHC) para estudar as aplicações de linguagens naturais em sistemas de informática. Dentre as principais aplicações compreendidas pelo PLN estão incluídas tradução automática, correção ortográfica, extração e recuperação de informações e sistemas de perguntas e respostas. Tais aplicações, que são voltadas para o usuário final, geralmente empregam uma cadeia de tarefas intermediárias para atingir o seu objetivo.

Em especial, a anotação de papeis semânticos (APS) ajuda a capturar o significado do texto, caracterizando os eventos descritos em suas sentenças através da identificação e anotação de estruturas predicado-argumento, podendo assim potencialmente beneficiar o desempenho dessas aplicações (MÀRQUEZ, *et al.*, 2008).

1.1. Anotação de papéis semânticos

"A análise semântica em nível sentencial de um texto se preocupa com a caracterização de eventos, tais como 'quem' fez 'o que' para 'quem', 'onde', 'quando' e 'como'. O predicado de uma frase (tipicamente um verbo) estabelece 'o que' aconteceu, e os outros componentes da frase expressam os participantes do evento (por exemplo 'quem' e 'onde'), assim como as demais propriedades do evento (por exemplo 'quando' e 'como'). A tarefa principal da ***anotação de papeis semânticos*** *(APS) é indicar exatamente quais relações semânticas existem entre o predicado e seus participantes e propriedades, selecionando essas relações a partir de uma lista pré-definida de possíveis* ***papeis semânticos*** *para esse predicado..."* (MÀRQUEZ, *et al.*, 2008).

A noção de papel semântico não é um conceito novo. Já em tempos datados de Antes de Cristo o gramático Pāṇini descreveu, na gramática de Sânscrito, as relações semânticas de dependência entre os substantivos e o verbo. Essas relações, denominadas kārakas, eram divididas em 6 categorias: kartṛ (agente), karman (objeto), karaṇa (instrumento), saṃpradāna (destino), apādāna (origem), adhikaraṇa (local) (BLAKE, 2001).

Em 1968 Charles J. Fillmore publica *Gramática de Casos*, uma obra de grande impacto na linguística computacional (JURAFSKY, 2004), trazendo como ideia central o fato de que a estrutura sintática de uma oração pode ser predita por seus participantes semânticos (FILLMORE, 1968). Nela, os casos (papéis semânticos) são generalizados os em 6 classes distintas: **Agentivo** (o causador da ação), **Instrumental** (força ou objeto envolvido na ação), **Dativo** (aquilo que é afetado pela ação), **Factitivo** (o resultado da ação), **Locativo** (localização) e **Objetivo** (caso neutro determinado a partir da interpretação do verbo).

Outra abordagem mais abrangente e menos generalista é proposta na *Semântica de Frames* (FILLMORE, 1985), onde se propõe que os significados das palavras podem ser melhor compreendidos através da evocação de janelas de conhecimento, conhecidas como *frames.* Ao analisar, por exemplo, o *frame* de conhecimento *transação comercial* teríamos elementos como comprador, vendedor, mercadoria, moeda de troca, etc.

Podemos realizar então a anotação de papéis semânticos usando qualquer uma dessas duas teorias. Vamos tomar como exemplo a frase "Pedro vendeu o carro ao

seu irmão no Rio de Janeiro". O verbo *vender* estabelece o evento ocorrido e assim representa o predicado da sentença. Os demais componentes são participantes ou propriedades desse predicado (argumentos). A classificação dos papeis semânticos de acordo com as duas teorias é mostrado na Tabela 1.

Argumento	Semântica de Frames	Gramática de Casos
Pedro	Vendedor	Agentivo
carro	Mercadoria	Objetivo
irmão	Comprador	Dativo
Rio de Janeiro	Lugar	Locativo

Tabela 1: Exemplo de etiquetas de papel semântico para Semântica de Frames e Gramática de Casos.

Para realizar a tarefa de APS de forma automatizada a grande maioria dos sistemas existentes emprega técnicas de aprendizado de máquina supervisionado, onde temos como entrada um corpus contendo uma imensa quantidade de sentenças anotadas com seus papéis semânticos (exemplos) que são processadas pelo sistema no intuito de aprender os padrões de atribuição de cada papel semântico (modelo). Em geral, quanto maior a quantidade de exemplos que um sistema dispõe na fase de treinamento melhor será seu resultado na hora de atribuir papéis semânticos a sentenças não vistas anteriormente (predição).

No entanto, ambas as teorias acima apresentam problemas ao tratar a tarefa de APS sob o enfoque do aprendizado de máquina. A gramática de casos, por empregar um conjunto genérico de papéis temáticos para todos os verbos, diversas vezes apresenta dificuldades sobre quais papéis devem ser atribuídos a cada um dos argumentos. Muitos trabalhos posteriores, como (MCCOY, 1969) e (NILSEN, 1972), propõe alterações ou extensões ao conjunto original de Fillmore porém sem chegar a um consenso comum (LIMA, 1982). Em contrapartida a semântica de frames define um conjunto diferente de papéis para cada grupo de verbos que fazem parte da mesma *janela de conhecimento*. Essa alta granularidade de papéis dificulta o aprendizado de máquina, exigindo recursos lexicais anotados maiores e por consequência mais caros de produzir.

O projeto PropBank (PALMER, KINGSBURY e GILDEA, 2005) foi criado com o intuito de fornecer dados que pudessem ser usados no treinamento de sistemas com base estatística e podemos dizer que se situa no meio do caminho

entre essas duas teorias. Ele adiciona uma camada com anotações de papéis semânticos sobre uma parte do Penn Tree Bank (MARCUS, SANTORINI e MARCINKIEWICZ, 1993), que por sua vez já possui árvores sintáticas anotadas para todas suas sentenças. É importante destacar que, na forma como foi idealizada, a anotação de papéis semânticos é derivada da anotação sintática, sendo assim sua dependente. Esse fator é de grande limitação na disponibilidade de recursos lexicais semanticamente anotados, já que geralmente são implementados como extensões aos corpora que já possuem anotação sintática prévia.

O modelo de anotação do PropBank ao invés de associar nomes aos argumentos principais dos verbos emprega rótulos numerados de **Arg0** a **Arg5**. O significado desses rótulos pode variar de verbo para verbo, mas geralmente Arg0 representa o agente prototípico e Arg1 o paciente prototípico. Além disso uma sentença pode não conter alguns desses rótulos. Em adição aos argumentos principais o PropBank também define um grupo com 14 argumentos modificadores comuns a todos os verbos, os **ArgM**. Esses argumentos são opcionais e compreendem expressões como local, tempo, causa, entre outros.

No Brasil o recurso lexical que segue essa linha é o Propbank.br (DURAN e ALUÍSIO, 2012). Assim como feito na versão americana, é adicionada uma camada de anotação semântica sobre um corpus já anotado sintaticamente, que nesse caso é a parcela Brasileira do corpus Bosque[1]. Diferente do que geralmente é praticado na produção desse tipo de material, a anotação do PropBank.Br não foi feita por uma equipe e sim por uma única linguista. Como consequência esse recurso possui apenas 6.142 proposições, o que é bem pouco quando comparado com as 112.917[2] proposições do PropBank americano (cerca de 18 vezes maior). Mais detalhes sobre sua composição e origem podem ser encontrados no Capítulo 3. De forma a contornar a limitação de tamanho, nos últimos anos, foram feitas tentativas de melhorar o desempenho da APS utilizando técnicas de aprendizado semisupervisionado (ALVA-MANCHEGO, 2013) ou não-supervisionado (FONSECA, 2013). Embora esses trabalhos tenham feito diversas contribuições

[1] O Bosque é composto de sentenças no estilo jornalístico extraídas dos jornais Folha de São Paulo (brasileiro) e Público (português) sendo parte integrante do projeto Floresta Sintá(c)tica (AFONSO, *et al.*, 2002).

[2] https://catalog.ldc.upenn.edu/docs/LDC2004T14/notes.txt

diretas e indiretas para a área de PLN, não foram capazes de superar o desempenho dos sistemas puramente supervisionados. Por esse motivo nesse trabalho iremos focar na tarefa de APS usando aprendizado supervisionado.

1.2. Support Vector Machines e Liblinear

Uma técnica de aprendizado supervisionado que vem se tornando bastante popular nos últimos anos e que tem atingido resultados excelentes é o SVM (*support vectors machine* ou máquina de vetores de suporte). Foi desenvolvida por (VAPNIK, 1995) com o intuito de resolver problemas de classificação de padrões. Nela exemplos de classes binárias são representados em planos n-dimensionais de forma a serem particionados por um hiperplano (um objeto de n-1 dimensões). Supondo que nossos exemplos fossem representados em um plano de duas dimensões nosso hiperplano seria um objeto de apenas uma dimensão, ou seja uma reta. A Figura 1 ilustra esse conceito.

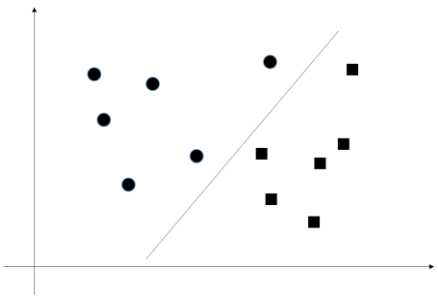

Figura 1: Exemplos em um plano de duas dimensões separados por uma reta.

Podemos dizer então que os exemplos acima são linearmente separáveis pelo hiperplano encontrado pela SVM. Porém problemas reais dificilmente mantém essa natureza, sendo a maioria deles complexos e não lineares. Para lidar com esses casos as SVM mapeiam os exemplos, através de funções Φ reais, em um espaço de maior dimensionalidade de forma a torná-los linearmente separáveis. Esse mecanismo é conhecido como *kernel trick* e o plano mapeado é denominado *espaço de características*. A Figura 2 ilustra esse processo.

De forma a obter uma melhor generalização do modelo as SVM buscam o *hiperplano ótimo* com a maior *margem* possível. Em outras palavras, coloca a maior quantidade possível de exemplos de uma classe em um dos lados ao mesmo tempo

em que maximiza a distância entre as duas classes. Na Figura 2 as margens são representadas por linhas pontilhadas.

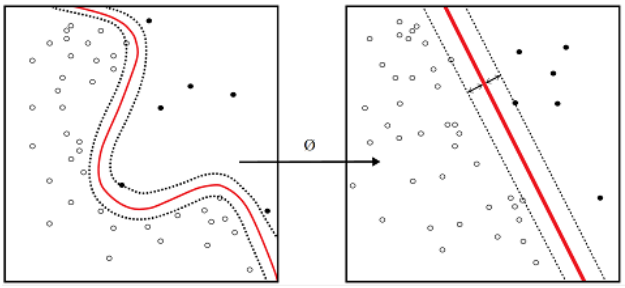

Figura 2: Mapeamento de exemplos em um espaço linearmente separável. Fonte: Wikipedia.

De forma a tratar o problema multiclasse, como é o caso da APS, foram desenvolvidas extensões ao modelo original da SVM. A metodologia dominante visa reduzir um problema multiclasse a diversos problemas de classificação binários (DUAN e KEERTHI, 2005). Alguns algoritmos que usam essa técnica são: *one-versus-all*, *one-versus-one* (HSU e LIN, 2002), SVM baseada em grafos direcionados acíclicos (PLATT, CRISTIANINI e SHAWE-TAYLOR, 2000) e códigos de saída com correção de erro (DIETTERICH e BAKIRI, 1995). Já (CRAMMER e SINGER, 2001) propõe resolver o problema de classificação multiclasse através de um único problema de otimização, ao invés de dividi-lo em várias partes menores.

No tocante a problemas de PLN uma ferramenta bastante conceituada em termos de desempenho e velocidade é a *liblinear* (FAN, *et al.*, 2008). Como é otimizada para tarefas de classificação em larga escala pode apresentar tempos de treinamento bem mais baixos que outras implementações baseadas em SVM para tarefas como a classificação de textos. São suportados classificadores baseados em regressão logística e SVM lineares e implementos os métodos *one-versus-all* e Crammer e Singer para problemas multiclasse. Nesse trabalho os modelos são construídos com o uso dessa ferramenta.

1.3. Seleção de Atributos

A seleção de atributos é uma técnica de regularização de domínio usada em aprendizado de máquina para escolher os atributos mais relevantes para a etapa de treinamento. Sua finalidade é ampla e pode ser eficientemente usada para reduzir o

consumo de memória ou tempo de treinamento assim como aumentar a generalização do modelo e a taxa de acertos na classificação (JAMES, *et al.*, 2012).

A ideia principal por trás dessa técnica é que os dados de entrada possuem muitos atributos que são irrelevantes ou redundantes, logo podem ser removidos sem incorrer em (muita) perda de informação. Note que esses dois conceitos são independentes: um atributo pode ser relevante, porém redundante em face de outro mais fortemente correlacionado (GUYON e ELISSEEFF, 2003) ao passo que um atributo irrelevante não agrega conhecimento ao modelo.

A estratégia mais comum para a seleção de atributos é chamada de *subset selection*, que consiste em escolher um subconjunto ótimo dos atributos disponíveis. Em sua forma mais simples o algoritmo avaliaria cada subconjunto possível e escolheria aquele com menor erro de classificação. Podemos calcular o número de subconjuntos de um conjunto *C* através da equação $N_s = 2^n$, onde *n* é o número de elementos de *C*. Imaginando um modelo com apenas 10 atributos de entrada esse algoritmo precisaria realizar 1024 comparações para chegar ao resultado. Percebe-se claramente que esse tipo de solução tende a se tornar computacionalmente intratável rapidamente.

Uma otimização popular dessa estratégia é o *greedy hill climbing*, que iterativamente modifica um subconjunto de atributos, um atributo por vez, e avalia se o novo subconjunto é melhor que o antigo. Nesse trabalho usaremos um algoritmo baseado nessa técnica para fazer a primeira camada de regularização.

Uma abordagem diferente para o problema de seleção de atributos é vista em (MOTTA, 2014), que propõe a seleção de atributos através de um *perceptron esparso multiclasse* modificado. A ideia central se baseia no princípio de que os atributos mais importantes estão envolvidos em mais atualizações do peso das classes. O algoritmo então mantém registro de quantas vezes cada um dos atributos participou de tais atualizações. Ao final escolhe-se os atributos que possuírem seu contador acima de determinado limiar. Note que essa estratégia realiza a seleção em nível binário ao passo que a anterior atua em nível categórico. A segunda camada de regularização desse trabalho usa uma adaptação desse conceito aplicado a SVM. Mais detalhes sobre a abordagem de Motta e sua adaptação estão disponíveis no capítulo 5.2.

1.4. Motivação e Objetivos

De forma geral, em *machine learning*, o objetivo mais importante de um modelo é atingir uma boa taxa de acerto em dados desconhecidos, ou seja, não vistos durante as fases de treino e desenvolvimento. Para atingir a essa meta o modelo precisa ser generalista o suficiente para não causar *overfitting*. Esse fenômeno ocorre quando o modelo começa a "memorizar" os dados de treinamento ao invés de "aprender" a generalizar os padrões de relacionamento dos atributos e tem alto risco de ocorrer em modelos muito complexos, onde o número de atributos é maior que o número de observações.

Esse problema é comum em sistemas de PLN, onde a maioria dos atributos é categórico e precisa ser convertido em um ou mais atributos numéricos para que possam ser processados por preditores de base estatística. Na grande maioria das vezes essa conversão se dá na forma binária, ou seja, cada instância de um atributo categórico se transforma um atributo binário no modelo final. Um exemplo disso é o atributo categórico *forma*, onde sua representação binária dá origem a mais de 12 mil atributos, um para cada palavra diferente presente no léxico (viajar, políticos, dinheiro, etc.).

Seguindo a linha de pesquisa do laboratório LEARN[3] nosso objetivo principal é construir um classificador que tenha o melhor desempenho possível em dados desconhecidos e para tanto o modelo precisa ser simples e compacto. Esse preditor precisa ter um poder de predição comparável aos atuais estado-da-arte.

1.5. Contribuições

Tendo em vista os objetivos citados no tópico anterior as principais contribuições desse trabalho são:

- A simplificação da tarefa de APS em uma única etapa, ao invés de separa-la em uma fase de identificação de atributos e uma segunda de classificação. Além disso eliminamos a poda como etapa de pré-processamento.

[3] Laboratório de Engenharia de Algoritmos e Redes Neurais (http://www.learn.inf.puc-rio.br)

- Desenvolveu-se um conjunto de 57 novos atributos (dos 114 usados nesse modelo) que usam diversas informações baseadas nas árvores sintagmáticas e de dependência. Vários desses em trabalho conjunto com uma especialista em linguística.
- Implementou-se um algoritmo baseado na técnica *greedy hill climbing* para realizar a seleção de atributos categóricos.
- Propôs-se uma metodologia de seleção de atributos inspirada no IFIS[4], porém voltada para modelos SVM, reduzindo a quantidade de atributos binários usados no modelo em 96%.
- A combinação dessas duas estratégias de regularização compactou o número de atributos binários para apenas 4% dos originais ao mesmo tempo em que elevou o desempenho em quase 2% de medida F.
- Por fim, criou-se um classificador de APS para o idioma português que usa o corpus PropBank.br e atinge desempenho ligeiramente superior ao estado-da-arte. A avaliação empírica é feita pelo script oficial da *CoNLL*[5] *2005 Shared Task* obtendo 82,17% de precisão, 82,88% de cobertura e 82,52% de F1 na classificação ao passo que o atual estado-da-arte (ALVA-MANCHEGO, 2013) relata 83,0% de precisão, 81,7% de cobertura e 82,3% de F1.

1.6. Organização da Dissertação

Os capítulos a seguir estão organizados da seguinte forma. No capítulo 2 nós revisitamos os principais trabalhos relacionados com a tarefa dando uma descrição sucinta de cada um deles. O capítulo 3 apresenta o conjunto de dados selecionado para a tarefa, detalhando sua metodologia de anotação e conjunto de rótulos. No capítulo 4 é apresentada nossa abordagem ao problema de APS, bem como os detalhes de modelagem da solução. Nos capítulos 5 detalhamos o mecanismo de regularização de atributos em duas camadas. O capítulo 6 relata os experimentos e

[4] Incremental Feature Induction and Selection (MOTTA, 2014)

[5] Conference on Natural Language Learning

resultados do trabalho usando a métrica oficial da *CoNLL 2005 Shared Task*. No capítulo 7 está a discussão dos resultados e conclusões finais.

2. Trabalhos relacionados

Em geral os sistemas que realizam a tarefa de APS automaticamente usam um de dois paradigmas: predição através do aprendizado de máquina ou codificação de regras. Os sistemas baseados em codificação de regras precisam prever e descrever todas as variações possíveis que um papel semântico pode assumir na sentença, muitas vezes se valendo de heurísticas para tal fim. Uma grande desvantagem desse tipo de abordagem é que as regras precisam ser criadas manualmente, o que exige grande esforço. Em contrapartida a ausência de dados anotados não afeta a eficiência desse tipo de classificador.

A segunda estratégia é construir classificadores baseados em aprendizado de máquina. Um sistema dessa categoria não precisa que as regras sejam codificadas explicitamente, uma vez que tem a capacidade de extrair os padrões de classificação dos argumentos a partir das observações contidas nos corpora que servem como dados de entrada. Considerando um exemplo do corpus de entrada como um elemento $x \in X$, onde x é representado pelo vetor de atributos $(x_1, x_2, \dots, x_n)$ e possui a etiqueta y, o objetivo do classificador é encontrar uma função $\hat{f}(x_1, x_2, \dots, x_n) = y$. Ou seja, definir um modelo onde se consiga mapear os atributos do exemplo x no rótulo y que o descreve. Nessa seção descreveremos os principais sistemas de publicados, em especial os destinados ao idioma português.

2.1. Sistemas de APS automática para Português

2.1.1. Eckhard Bick (2007)

O parser "Palavras" (BICK, 2002) é um sistema modular baseado em codificação de regras, sendo o maior representante desse paradigma para o português. O módulo que realiza a tarefa de APS foi proposto em (BICK, 2007) e possui um conjunto de 500 regras escritas manualmente para atribuir uma dentre as 38 classes de papel semântico inspiradas nos trabalhos (HAJICOVA, PANEVOVA e SGALL, 2000) e (TAULÉ, *et al.*, 2005). O desempenho é testado sobre um

conjunto de 2.500 palavras extraídas da parte europeia do Floresta Sintá(c)tica[6] que foi previamente anotado e manualmente revisado com adição de informações de dependência, protótipo semântico (BICK, 2006) e entidades mencionadas, atingindo 88.6% de F1.

2.1.2. Amancio, Duran e Aluisio (2010)

O estudo proposto por (AMANCIO, DURAN e ALUISIO, 2010) compara diferentes algoritmos para a tarefa de APS automática em português obtendo o melhor resultado com SMO[7], onde atinge 79 de medida F1. O conjunto de papeis semânticos é baseado em perguntas contendo etiquetas do tipo quem, por que, onde, quando, etc. Foi utilizado para a tarefa o *corpus* do projeto Por-Simples (CASELI, *et al.*, 2009) contendo 104 notícias simplificadas manualmente que foram extraídas do jornal brasileiro Zero Hora. A esse corpus foram adicionadas: uma camada de anotação sintática realizada pelo *parser* Palavras sem revisão manual e uma camada de anotação semântica contendo as etiquetas de pergunta, adicionada manualmente.

O sistema usa um conjunto composto de 13 grupos de atributos tradicionais, totalizando 23 atributos distintos, a seguir: *Phrase type, Side, Argument order, Subcategorization of syntactic functions, Specific syntactic function, Question at the Left side, Number of arguments, Principal verb token, First two Part Of Speech (POS) and Last POS of an argument, First and Second tokens of an argument, Semantic values of the argument tokens, Simple or Multiword verb e Number of tokens of the argument.*

2.1.3. Sequeira, Gonçalves e Quaresma (2012)

O sistema implementado por (SEQUEIRA, GONÇALVES e QUARESMA, 2012) realiza a tarefa de APS para português europeu obtendo o melhor resultado de 31.1 de F1 na classificação de ARG0 e 19.0 em ARG1 quando usado o modelo

[6] O projeto Floresta Sintá(c)tica é uma colaboração entre a Linguateca e o projecto VISL. Contém textos em português (do Brasil e de Portugal) anotados automaticamente pelo analisador sintático PALAVRAS e revistos por linguistas (http://www.linguateca.pt/floresta).

[7] Sequential Minimal Optimization (SMO) é um algoritmo para solucionar o problema de programação quadrática (QP) que surge durante o treinamento das SVM (PLATT, 1998).

SVM. Usou-se um subconjunto minimalista das etiquetas do PropBank contendo apenas as etiquetas P (Predicado), ARG0 (agente prototípico) e ARG1 (paciente prototípico). Foi utilizada para a tarefa seção CETEMPúblico (ROCHA e SANTOS, 2000) do *corpus* Bosque[8] contendo um total de 4.416 sentenças. A camada de papéis semânticos foi adicionada automaticamente e sem revisão manual, contendo apenas as etiquetas P, ARG0 e ARG1.

O artigo não descreve detalhadamente os atributos incluídos mencionando apenas o uso de uma janela contextual de 3 palavras.

2.1.4. Alva-Manchego (2013)

A dissertação de mestrado (ALVA-MANCHEGO, 2013) desenvolve vários classificadores para a tarefa de APS automática em português brasileiro. Seu melhor resultado fica por conta do sistema supervisionado que usa dois classificadores baseados no algoritmo de máxima entropia (FLEISCHMAN, KWON e HOVY, 2003), um na identificação de argumentos e outro na classificação dos argumentos previamente identificados, a tarefa combinada atinge 82,3 de F1. Usa-se como conjunto de papeis semânticos as etiquetas do projeto PropBank.

O *corpus* escolhido foi o PropBank.Br que, após passar por uma etapa de conversão para o formato plano por colunas usado na *CoNLL 2005 Shared Task*, teve 1.331 sentenças descartadas por falhas nessa conversão, mantendo um total de 3.308 sentenças anotadas com papéis semânticos *gold-standard*. Uma segunda conversão é feita na camada de árvores de dependência, onde as relações que originalmente foram definidas de constituinte para constituinte são transformadas em de palavra para palavra. Ao final as anotações de papel semântico são transferidas da árvore sintagmática para a de dependência e todo o processo é revisado de forma semiautomática. Esse procedimento de conversão resultou no *corpus* PropBank.Br 1.1[9] que é disponibilizado no portal da Universidade. Mais detalhes sobre sua implementação e composição podem ser vistos no capítulo 3.

[8] O corpus Bosque faz parte do projeto Floresta Sintá(c)tica (SANTO, BICK e AFONSO, 2007) e é composto pelos primeiros 1.000 extratos dos corpora CETENFolha e CETEMPúblico.

[9] http://www.nilc.icmc.usp.br/portlex/index.php/pt/projetos/propbankbr

O sistema implementa um conjunto de 57 atributos encontrados na literatura e realiza uma seleção baseada na estratégia *greed hill climbing*, para chegar a um subconjunto de apenas 16 atributos que são usados no classificador supervisionado. Além disso o sistema incorpora posteriormente atributos de dependência que não passam pelo mecanismo de seleção. O conjunto final possui 17 atributos, que são: Primeira Palavra + POS 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, Lema do Predicado e Função Sintática.

2.1.5. Fonseca (2013)

A dissertação de mestrado (FONSECA, 2013) propõe um classificador baseado em *Deep Learning*[10] para realizar a tarefa obtendo 68,0 de medida F1. O conjunto de etiquetas de papel semântico é o mesmo proposto pelo projeto PropBank. O corpus selecionado é o PropBank.Br contendo as 3.308 sentenças que foram usadas no trabalho de (ALVA-MANCHEGO, 2013). O sistema usa técnicas de aprendizado não-supervisionado, logo o conjunto de atributos é totalmente induzido e não legível por humanos.

2.2. Sistemas de APS automática para outros idiomas

2.2.1. Gildea e Jurafsky (2002)

O anotador proposto por (GILDEA e JURAFSKY, 2002) foi o pioneiro no uso de métodos de aprendizagem estatística na realização da tarefa de APS. Ele emprega um modelo chamado de *backoff lattice*[11] e obtém a marca de 64,3 de

[10] Deep Learning compreende um conjunto de algoritmos que modelam abstrações de dados de alto nível usando um grafo com várias camadas de processamento onde são executadas transformações lineares e não lineares (GOODFELLOW, BENGIO e COURVILLE, 2016).

[11] *Backoff lattice* executa a predição através da busca pelo exemplo que contenha o maior número possível de atributos com os mesmos valores da instância a ser predita.

medida F1 realizando a tarefa em duas etapas: identificação dos argumentos e posterior classificação desses argumentos. O conjunto de etiquetas de papel semântico, conhecidas como *frame elements*, são as definidas pelo projeto FrameNet (BAKER, FILLMORE e LOWE, 1998). O *corpus*, originário do mesmo projeto, contém 50.000 sentenças anotadas manualmente com esse tipo de etiqueta.

Uma das contribuições mais relevantes desse estudo foi o conjunto de atributos desenvolvido, que passou a sem usado amplamente por trabalhos subsequentes. A seguir listamos esses atributos: Tipo de Sintagma, Categoria Principal (indica se um determinado sintagma nominal é sujeito ou objeto direto do verbo), Caminho na Árvore Sintagmática, Posição, Voz, Núcleo do Sintagma e Subcategorização.

2.2.2. Pradhan, Ward e Martin (2008)

O estudo produzido por (PRADHAN, WARD e MARTIN, 2008) compara o desempenho de sistemas de APS automática, em inglês, quando treinados e testados com *corpora* de gêneros diferentes. O preditor usado constrói um modelo SVM multiclasse usando a estratégia *one-versus-all* e consegue 80,0 de F1 nos experimentos com o *corpus* PropBank e 63,9 no *corpus* Brown. O conjunto de etiquetas de papel semântico é o constante no projeto PropBank. São usados dois *corpora* para a tarefa, ambos anotados conforme as definições do PropBank, o primeiro é o próprio PropBank (baseado no *Wall Street Journal*) e o segundo é o Brown (que contém textos de 15 categorias diferentes).

Os atributos usados são: Forma e Lema do Verbo, Caminho, Tipo de Sintagma, Posição, Voz, Subcategorização, Núcleo do Sintagma, *Cluster* do Verbo, POS do Núcleo, Entidade Nomeada no Constituinte, Caminho parcial, Caminho de frases, caminho de n-gramas, caminho de tipo de sintagma de um caractere, Contexto do verbo (janela de tamanho 2), Pontuação à esquerda e à direita, Núcleo do Sintagma Preposicional, Primeira e Última Palavra/POS no Constituinte, Posição Ordinal do Constituinte, Distância em Constituintes na Árvore, Atributos dos Parentes do Constituinte, Palavras Temporais, Frame Sintático.

2.2.3. Toutanova, Haghighi e Manning (2008)

O sistema proposto em (TOUTANOVA, HAGHIGHI e MANNING, 2008) define um argumento semântico como uma estrutura composta que se relaciona fortemente com outros argumentos semânticos da mesma sentença. Ou seja, os atributos de entrada e etiqueta selecionada para um determinado argumento do verbo influenciam na atribuição de papel semântico dos demais argumentos desse mesmo verbo, mudando o paradigma de atribuição das etiquetas de individual para grupal. O classificador implementa uma abordagem de *re-ranking* logaritmo linear (COLLINS e KOO, 2005) para realizar a anotação de todas as etiquetas semânticas da sentença em conjunto, obtendo 91,2 de F1 com árvores sintáticas *gold-standard* e 80,0 nas automáticas. Como de costume o sistema também é dividido em uma etapa de identificação e outra de classificação. O conjunto de etiquetas semânticas e o corpus são os do projeto PropBank.

Atributos: Tipo de Sintagma, Lema do Verbo, Caminho, Posição, Voz, Núcleo, Subcategorização, Primeira e Última Palavra do Constituinte, Atributos dos Parentes do Verbo, Caminho Parcial, Núcleo do Sintagma Preposicional, Núcleo do Pai do Sintagma Preposicional, Lema do Verbo + Caminho, Lema do Verbo + Núcleo, Lema do Verbo + Tipo de Sintagma, Voz + Posição e Lema do Verbo + Núcleo do Pai do Sintagma Preposicional, Sujeito Ausente, Caminho Projetado.

2.2.4. Morante e Bosch (2009)

Em (MORANTE e BOSCH, 2009) é proposto um classificador baseado em memória, realizando a tarefa de APS nas corriqueiras fases de identificação e classificação, atingindo 88,9 de F1 para catalão dentro do domínio, 85,3 de F1 para catalão fora do domínio, 84,0 de F1 para espanhol dentro do domínio e 87,4 de F1 para espanhol fora do domínio. O conjunto de etiquetas semânticas é derivado do PropBank e os *corpora* são os disponibilizados na SemEval-2007, contendo 3.611 sentenças para espanhol e 3.202 sentenças para catalão, ambos com anotação de árvore sintática *gold*. Detalhes sobre as etiquetas e a composição dos corpora podem ser encontrados em (MÁRQUEZ, *et al.*, 2007).

Foi usado no estudo um conjunto de 323 atributos descritos em (GILDEA e JURAFSKY, 2002), (XUE e PALMER, 2004), (CARRERAS e MÁRQUEZ,

2004), (CARRERAS e MÁRQUEZ, 2005) e (TOUTANOVA, HAGHIGHI e MANNING, 2005). Foi feita uma seleção baseada em *greed hill climbing* resultando num conjunto final de 88 atributos.

2.3. Conclusões do capítulo

Nesse capítulo apresentamos os sistemas de APS automática baseados em aprendizado de máquina que consideramos os mais relevantes em relação ao contexto desse trabalho, tanto no idioma português quanto em outros idiomas. Foram descritas as diferentes abordagens usadas em cada um deles bem como os principais atributos desenvolvidos pelos sistemas que usam aprendizado supervisionado.

É fácil de se notar que a quantidade de esforços dedicados ao idioma inglês é muito mais abundante que em qualquer outro idioma, em particular o português, e que isso com certeza é uma das principais razões para a diferença no tamanho e variedade dos corpora disponíveis.

3. Conjunto de dados

O *corpus* escolhido para a tarefa é o PropBank.Br 1.1 (ALVA-MANCHEGO, 2013) que foi elaborado de acordo com as definições da *CoNLL 2005 Shared Task*, agregando ao sistema alguns benefícios, tais como: conjuntos de dados de treinamento e teste propriamente divididos e anotados ao estilo PropBank, possibilidade de avaliação pelo script oficial da competição e formato de arquivo plano por colunas equivalentes aos originais da competição. Além disso, a escolha de uma configuração padronizada[12], permite que os resultados desse trabalho sejam empiricamente comparáveis com outros estudos baseados no mesmo padrão.

O PropBank.Br 1.1 é derivado do PropBank.Br (DURAN e ALUÍSIO, 2012) que foi construído sobre a porção de textos brasileiros do *corpus* Bosque 8.0. O Bosque por sua vez é um *subcorpus* do Floresta Sintá(c)tica (AFONSO, *et al.*, 2002) que contém 4.213 sentenças extraídas do jornal brasileiro Folha de São Paulo e 5.224 extraídas do jornal português Público. Por ser um *Treebank gold-standard*, todas as sentenças do Bosque possuem anotações de árvore sintática manualmente revisadas por linguistas. As sentenças do PropBank.Br possuem anotação de papel semântico *gold-standard* que seguem as definições do projeto PropBank americano.

O PropBank.Br (DURAN e ALUÍSIO, 2012) é disponibilizado em um único arquivo no formato Tiger-XML sem divisão entre conjuntos de treinamento e teste. De forma a adequá-lo às definições da *CoNLL 2005 Shared Task* (ALVA-MANCHEGO, 2013) primeiramente realizou um processo de conversão para o formato plano por colunas utilizado na competição. Durante esse processo 1.331 sentenças[13] foram descartadas por problemas variados.

[12] Essa configuração é composta por: conjuntos de treinamento e teste, formato dos arquivos de saída e metodologia de avaliação.

[13] 312 por algum tipo de erro que atrapalhou a anotação manual de papéis semânticos, 16 por erros na árvore sintática, 964 por serem instâncias do verbo ser, 25 por estrutura incompleta de argumentos e 14 por argumentos aninhados.

O segundo passo seguido por (ALVA-MANCHEGO, 2013) para tornar o *corpus* PropBank.Br compatível com a *Shared Task* foi separa-lo em conjuntos de treinamento e teste. “Para realizar essa 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 conjuntos de dados. As novas sentenças que aparecem no *corpus* PropBank.Br (versão 8.0 do Bosque), foram adicionadas ao conjunto de teste”.

O terceiro e último passo executado por (ALVA-MANCHEGO, 2013) foi a inclusão de anotação sintática de dependência. Essa informação se encontra disponível no arquivo TigerXML do PropBank.Br, no entanto, as relações de dependência estão estabelecidas entre constituintes e não entre palavras. Foi desenvolvido então um script automatizado com objetivo de extrair os núcleos dos constituintes e converter as relações para o segundo formato. Em seguida realizou-se a transferência das anotações de papel semântico das árvores de constituintes para as árvores de dependência. Para isso, empregou-se o método de (SURDEANU, *et al.*, 2008) usado no PropBank, no qual o papel semântico é atribuído ao núcleo do constituinte. Finalmente foi realizada uma etapa de verificação semiautomática conferindo status *gold-standard* às anotações de dependência.

Além das anotações de papel semântico e das árvores sintáticas sintagmáticas e de dependência o PropBank.Br 1.1 incluí outras anotações que se encontram detalhadas nas seções seguintes.

3.1. Composição do corpus

Cada sentença no *corpus* PropBank.Br 1.1 pode ter um ou mais predicativos-alvo, e para cada predicativo-alvo é anotado seu conjunto de argumentos semânticos. A esse grupamento do predicativo-alvo e seus argumentos damos o nome de proposição. As proposições no *corpus* já se encontram normalizadas, sendo assim cada proposição é representada como uma cópia da sentença original, trazendo consigo todas as suas anotações, com exceção do verbo-alvo e argumentos que mudam de proposição para proposição. No exemplo abaixo a sentença de dois

verbos (1.a.) é representada como duas proposições diferentes (1.b. e 1.c.) no *corpus* normalizado.

1.a. Aqui só joga quem está bem.

1.b. [Aqui **ARG**] [só **ARG**] [joga **V**] [quem está bem **ARG**].

1.c. Aqui só joga [quem **ARG**] [está **V**] [bem **ARG**].

O PropBank.Br 1.1 vem dividido em dois arquivos, um contendo as anotações de árvore sintagmática e outro contendo as anotações de dependência. Para esse projeto vamos precisar de ambas as anotações e foi feita uma unificação desses arquivos. O conjunto final de colunas que representam as anotações de uma proposição é apresentada em Tabela 2. Em seguida apresentamos uma proposição completamente anotada na Figura 3.

Núm.	Nome	Descrição
1	ID	Contador de *tokens* que inicia em 1 para cada nova proposição
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	MORF	Atributos morfológicos *gold-standard*
6	DTREE	Árvore de dependência *gold-standard*
7	FUNC	Função sintática do *token* para com seu regente na árvore de dependência
8	CTREE	Árvore sintagmática *gold-standard* completa
9	PRED	Predicados semânticos na proposição
10	ARG	Papel semântico do regente do argumento na árvore de dependência, conforme notação do PropBank

Tabela 2: Anotação de cada coluna de uma proposição.

ID	FORM	LEMMA	GPOS	MORF	DTREE	FUNC	CTREE	PRED	ARG
1	O	o	art	M|S	2	>N	(FCL(CU(NP*	-	*
2	grupo	grupo	n	M|S	7	SUBJ	*)	-	A0
3	e	e	conj-c	-	2	CO	*	-	*
4	o	o	art	M|S	5	>N	(NP*	-	*
5	governo	governo	n	M|S	2	CJT	*	-	*
6	iraniano	iraniano	adj	M|S	5	N<	(ADJP*)))	-	*
7	negaram	negar	v-fin	PS/MQP|3P|IND	0	STA	(VP*)	negar	*
8	envolvimento	envolvimento	n	M|S	7	ACC	(NP*)	-	A1
9	.	.	pu	-	7	PU	*)	-	*

Figura 3: Proposição completamente anotada.

Durante o desenvolvimento desse projeto foram percebidos alguns pequenos erros no conjunto de treinamento, a seguir:

- O *token* da linha 45.810, com FORMA=existente, exibia erradamente POS=VP que não faz parte do conjunto de etiquetas para essa coluna. Também nesse *token* "V|PCP|F|S" aparecia incorretamente na coluna MORF. Ajustamos POS para "V-PCP" e MORF para "F|S";
- O *token* da linha 5.143, com FORMA=peemedebistas, teve MORF corrigido de "M|R" para "M|P";
- Os tokens das linhas 50.112 e 96.521, FORMA=anteontem, possuem as colunas LEMA="xx", POS="xx", MORF="-". Nesse caso não fizemos correção alguma.

3.2. Anotações do corpus

Nessa seção estão descritas em maiores detalhes algumas das anotações da Tabela 2 que consideramos necessitar de maiores esclarecimentos.

3.2.1. Classe gramatical *gold-standard* (GPOS)

Classe gramatical ou classe de palavra é o nome dado ao conjunto que classifica uma palavra, baseando-se na sua distribuição sintática e morfológica. Na gramática da língua portuguesa a classificação tradicional estipula onze desses conjuntos (MESQUITA e MARTOS, 1994). O **substantivo** e o **verbo** geralmente são os elementos centrais de uma frase e são considerados a base para as demais relações. **Artigo**, **numeral**, **pronome**, **adjetivo** e **advérbio** geralmente qualificam substantivos e verbos e são consideradas *classes adjuntas*. **Preposição**, **conjunção**

e **pronome** expressão ideias de ligação e são consideradas *classes conectivas*. Já a **interjeição** serve para expressar emoções, sentimentos e sensações. As etiquetas usadas pelo corpus Bosque possuem um grau de detalhamento maior e estão listadas na Tabela 3.

Classe	Etiqueta
Substantivos	n
Substantivos/Adjetivos	n-adj
Adjetivos	adj
Nomes próprios	prop
Advérbios	adv
Verbos finitos	v-fin
Verbos gerúndios	v-ger
Verbos particípios	v-pcp
Verbos infinitivos	v-inf
Artigos	art
Pronomes determinativos	pron-det
Pronomes relativos	pron-rel
Pronomes pessoais	pron-pess
Advérbios	adv
Preposições	prp
Interjeições	intj
Conjunções subordinativas	conj-s
Conjunções coordenativas	conj-c
Prefixos	ec
Pontuação	pu

Tabela 3: Conjunto de etiquetas POS usadas pelo corpus Bosque.

3.2.2. Atributos morfossintáticos *gold-standard* (MORF)

Essa coluna é um campo multivalorado que incluí uma série de informações morfossintáticas associadas a **forma** ou **lema** do token. As etiquetas estão separadas por uma barra vertical (|) e abrangem classificações de género, número, caso, pessoa, tempo, modo, etc. A lista completa pode ser vista na Tabela 4.

Descrição	Etiqueta
Sujeito (incluindo sujeitos impessoais se)	SUBJ
Objeto direto (incluindo alguns tipos de se)	ACC
Objeto indireto pronominal (incluindo se)	DAT
Objeto preposicional	PIV
Objeto nominal	NOM
Objeto preposicional nominal	NOM/PIV
Complemento nominal	N<ARG
Singular	S
Plural	P
Singular/Plural	S/P
Masculino	M
Feminino	F
Masculino/Feminino	M/F
Outras categorias gramaticais usadas com propriedades nominais	N
Primeira pessoa do singular	1S
Segunda pessoa do singular	2S
Terceira pessoa do singular	3S
Primeira pessoa do plural	1P
Segunda pessoa do plural	2P
Terceira pessoa do plural	3P
Primeira/Terceira pessoa do singular	1/3S
Terceira pessoa do singular/plural	3S/P
Condicional	COND
Imperativo	IMP
Imperfeito	IMPF
Indicativo	IND
Futuro	FUT
Mais que perfeito	MQP
Presente	PR
Passado	PS
Passado/Mais que perfeito	PS/MQP

Tabela 4: Conjunto de etiquetas morfossintáticas do Propbank.Br.

3.2.3. Árvore sintática de dependência *gold-standard* (DTREE)

Essa coluna guarda o ID do regente do *token* atual conforme a árvore de dependência. No caso de apresentar valor 0 (zero) significa que esse elemento se liga diretamente a raiz virtual da sentença. No caso do PropBank.Br cada sentença só possui um *token* que se liga a raiz. A Figura 4 ilustra a árvore de dependência montada a partir das informações da coluna DTREE da Figura 3.

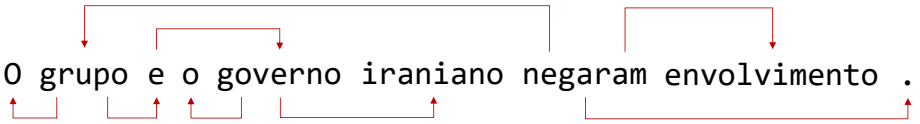

Figura 4: Exemplo de uma árvore de dependência gerada pela anotação de DTREE.

3.2.4. Função sintática (FUNC)

Esse atributo expressa a relação de dependência entre o *token* e seu regente especificado na coluna DTREE. A Tabela 5 contém uma listagem com todas etiquetas desse tipo encontradas no corpus PropBank.Br 1.1. A coluna descrição foi alimentada de acordo com a documentação da Floresta Sintá(c)tica[14].

Descrição	Etiqueta
Dependente à esquerda de um núcleo de natureza adjetival	>A
Dependente à esquerda de um núcleo de natureza nominal	>N
Dependente à esquerda de uma preposição e seu dependente	>P
Dependente à direita de um núcleo de natureza adverbial	A<
A ser definida	A<ARG
Objeto direto	ACC
Construção passiva para objeto direto	ACC-PASS
Adjunto adverbial	ADVL
Aposição do substantivo	APP
Verbo auxiliar	AUX

[14] http://linguateca.pt/Floresta/programas/FS/metadados/descEtiquetas.txt e http://www.linguateca.pt/Floresta/BibliaFlorestal/sec06.html

Ligação entre o auxiliar e os verbos coordenados	AUX<
Conjunto	CJT
Coordenador	CO
Enunciado imperativo	COM
Objeto indireto pronominal	DAT
Enunciado exclamativo	EXC
Marcador de foco	FOC
Núcleo	H
Complemento comparativo	KOMP<
Verbo principal	MV
Dependente à direita de um núcleo de natureza nominal	N<
Complemento nominal (complementa um substantivo não deverbal)	N<ARG
Argumento à direita de um núcleo de natureza nominal (Complemento Nominal que corresponde ao objeto de nomes deverbais)	N<ARGO
Argumento à direita de um núcleo de natureza nominal (Complemento Nominal que corresponde ao sujeito de nomes deverbais)	N<ARGS
Adjeto predicativo (dependente do núcleo por adição de informação)	N<PRED
Dependente de numeral	NUM<
Complemento adverbial	OA
Predicativo do objeto	OC
Predicador	P
Dependente à direita de um núcleo que é uma preposição	P<
Agente da passiva	PASS
Conjunto de preposições	PCJT
Objeto preposicional	PIV
Adjunto predicativo	PRED
Partícula de ligação verbal	PRT-AUX
Pontuação	PU
Enunciado interrogativo	QUE
Aposto da oração	S<
Complemento adverbial	SA
Predicativo do sujeito	SC
Enunciado declarativo	STA
Subordinador	SUB

Sujeito (incluindo sujeitos impessoais se)	SUBJ
Constituinte de tópico	TOP
Enunciado	UTT
Constituinte vocativo	VOC
Etiqueta não documentada	X

Tabela 5: Conjunto de etiquetas FUNC do PropBank.Br 1.1.

3.2.5. Árvore sintática de constituintes *gold-standard* (CTREE)

Essa coluna guarda a árvore sintática de constituintes completa no formato plano usado pela *CoNLL*. Os parêntesis indicam o início e fim dos sintagmas, que podem ser aninhados, e os asteriscos indicam a posição em que o *token* deve ficar na árvore. Para melhor entendimento inclui-se a árvore da Figura 5 construída a partir da coluna CTREE da Figura 3.

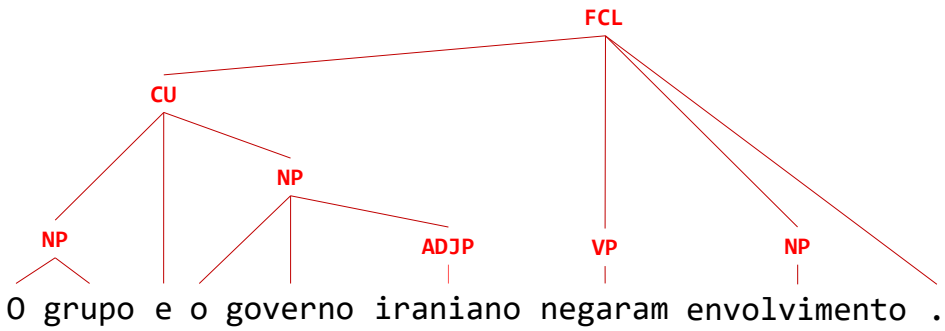

Figura 5: Exemplo de árvore sintática de constituintes gerada através da coluna CTREE.

3.2.6. Papel semântico do argumento (ARG)

O PropBank.Br 1.1 segue o modelo de anotação do PropBank que define um conjunto de argumentos para cada verbo individualmente e não atribui nomes aos mesmos, utilizando, em vez disso, rótulos numerados de A0 a A5. Cada verbo determina a quantidade de argumentos que podem acompanhá-lo e seus significados. Além disso, o PropBank define um conjunto de argumentos opcionais comuns a todos os verbos, os ArgM, que são expressões indicando tempo, local, modo, etc. A Tabela 6 apresenta uma lista com todos os argumentos do PropBank.Br.

Etiqueta	Descrição
A0	Tipicamente o agente prototípico
A1	Tipicamente o paciente prototípico ou tema
A2...A5	Significado dependente do verbo
AM-ADV	Modificador adverbial
AM-CAU	Causa
AM-DIR	Direção
AM-DIS	Marcador discursivo
AM-EXT	Extensão
AM-MED	Etiqueta não documentada
AM-LOC	Lugar
AM-MNR	Maneira
AM-NEG	Negação
AM-PNC	Propósito
AM-PRD	Predicação secundária
AM-REC	Recíproco, referenciando outro argumento
AM-TMP	Temporal

Tabela 6: Conjunto de etiquetas de papel semântico do PropBank.Br 1.1

No PropBankBr 1.1 com anotações de dependência a coluna ARG traz a etiqueta de papel semântico anotada apenas na raiz da subárvore que define o argumento. Isso significa dizer que todos os nós que fazem parte dessa subárvore compõem o argumento rotulado por ARG. Usando como referência a proposição da Figura 3 e a árvore de dependência da Figura 4 teríamos a seguinte anotação de argumentos na forma plana:

[O grupo e o governo iraniano $_{\mathbf{A0}}$] negaram [envolvimento $_{\mathbf{A1}}$].

3.3. Conjuntos de Treinamento, Validação e Teste

Conforme descrito na seção 3, uma das vantagens em se trabalhar com o *corpus* PropBank.Br 1.1 é que este já se encontra propriamente dividido em conjuntos de treinamento e teste, mantendo assim a compatibilidade com as definições da *CoNLL 2005 Shared Task*. Essa divisão mantém certa de 95,5% das sentenças no conjunto de treinamento e 4,5% em teste.

Para este trabalho realizamos a divisão do conjunto de treinamento em duas partes: desenvolvimento e validação. Essa divisão tem o intuito de manter o conjunto de teste inacessível até o momento dos testes finais. Todas as verificações e checagens nesse projeto foram feitas através da avaliação dos conjuntos de desenvolvimento/validação ou de validação cruzada no conjunto de treinamento. A construção do conjunto de validação seguiu os seguintes passos:

1. Ordena-se aleatoriamente as proposições do conjunto de treinamento.
2. Seleciona-se de 1 em 1 as proposições desse conjunto.
3. Para cada proposição selecionada, identificamos a sentença S da qual ela faz parte e incluímos no conjunto de validação **todas as demais** proposições desta sentença S.
4. Repetem-se os passos 2 e 3 até que sejam selecionadas pelo menos 10% das proposições.

O conjunto de validação gerado por esse procedimento contém 10% das proposições e 7,7% das sentenças e mantem aproximadamente a mesma distribuição de papéis semânticos do conjunto de treinamento. O gráfico da Figura 6 nos permite visualizar rapidamente como ficou a proporção final da distribuição das proposições entre os conjuntos de dados. Na próxima seção são detalhados maiores aspectos dessa divisão.

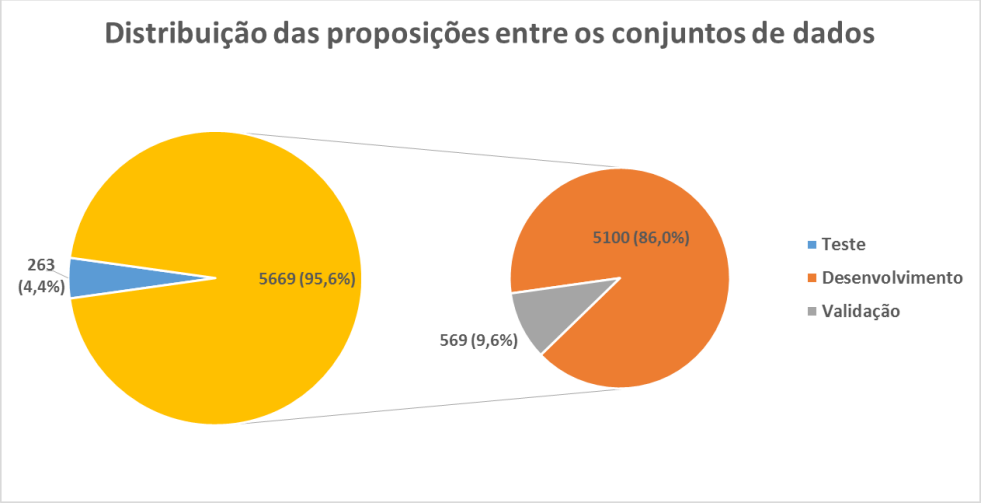

Figura 6: Gráfico de distribuição das proposições entre os conjuntos de dados

3.4. Análise do corpus

Apesar de pequeno o PropBank.Br 1.1 possui exemplos suficientes para que se possam iniciar pesquisas de aprendizado de máquina supervisionado preliminares, no entanto, conforme mencionado no capítulo 7.3, para realmente evoluir nessa linha é desejável que haja uma expansão da base de conhecimento. A distribuição das dos argumentos semânticos no conjunto de treinamento é desbalanceada, sendo formada em sua vasta maioria das etiquetas A0 e A1 que somam quase 60%. As etiquetas A2, AM-LOC e AM-TMP ainda possuem alguma representatividade somando cerca de 23%. As 13 restantes são bastante esparsas e somam cerca de 13% do total de argumentos semânticos.

Se não fosse o tamanho reduzido do *corpus* essa não seria uma característica indesejável, já que problemas do mundo real tendem a apresentar uma distribuição semelhante a essa. A maior dificuldade aqui é o correto aprendizado de etiquetas que quase não possuem instâncias no conjunto de treinamento. Etiquetas como A5, AM-MED e AM-REC possuem tão poucas instâncias que não conseguiram ser selecionadas pelo procedimento de criação do conjunto de validação. Já as etiquetas AM-DIR e AM-EXT que possuem respectivamente 12 e 77 instâncias dificilmente serão aprendidas pelo classificador. A Tabela 7 e a Figura 7 mostram um comparativo das estatísticas de distribuição nos conjuntos de treinamento, desenvolvimento e validação.

	Treinamento		Desenvolvimento		Validação	
Proposições	5.669	100%	5.100	90,0%	569	10,0%
Sentenças	3.208	100%	2.960	92,3%	248	7,7%
Tokens	136.060	100%	120.928	88,9%	15.132	11,1%
Verbos Distintos	1.004	100%	960	95,6%	294	29,3%
Argumentos	13.495	100%	12.177	90,2%	1.318	9,8%
A0	2.924	21,7%	2.619	21,5%	305	23,1%
A1	5.077	37,6%	4.566	37,5%	511	38,8%
A2	1.319	9,8%	1.190	9,8%	129	9,8%
A3	145	1,1%	135	1,1%	10	0,8%
A4	109	0,8%	101	0,8%	8	0,6%

A5	1	0,0%	1	0,0%	0	0,0%
AM-ADV	353	2,6%	316	2,6%	37	2,8%
AM-CAU	150	1,1%	136	1,1%	14	1,1%
AM-DIR	12	0,1%	11	0,1%	1	0,1%
AM-DIS	298	2,2%	280	2,3%	18	1,4%
AM-EXT	77	0,6%	67	0,6%	10	0,8%
AM-LOC	683	5,1%	627	5,1%	56	4,2%
AM-MED	3	0,0%	3	0,0%	0	0,0%
AM-MNR	389	2,9%	358	2,9%	31	2,4%
AM-NEG	320	2,4%	287	2,4%	33	2,5%
AM-PNC	149	1,1%	124	1,0%	25	1,9%
AM-PRD	177	1,3%	162	1,3%	15	1,1%
AM-REC	8	0,1%	8	0,1%	0	0,0%
AM-TMP	1.105	8,2%	1.009	8,3%	96	7,3%

Tabela 7: Estatísticas dos conjuntos de treinamento, desenvolvimento e validação.

Os conjuntos de desenvolvimento e validação são subdivisões do conjunto de treinamento, logo, os valores percentuais para as 5 primeiras linhas da tabela são expressos como frações em relação a esse conjunto. Por exemplo o conjunto de treinamento possui 3.208 sentenças no total (100%) enquanto que o conjunto de validação incluí apenas 248 dessas 3.208 sentenças, ou seja, 7,7%. Já da linha 6 em diante os percentuais são calculados em relação ao total de argumentos do próprio conjunto. Como exemplo temos validação com um total de 1.318 argumentos onde destes 305 são A0, logo 23,1%.

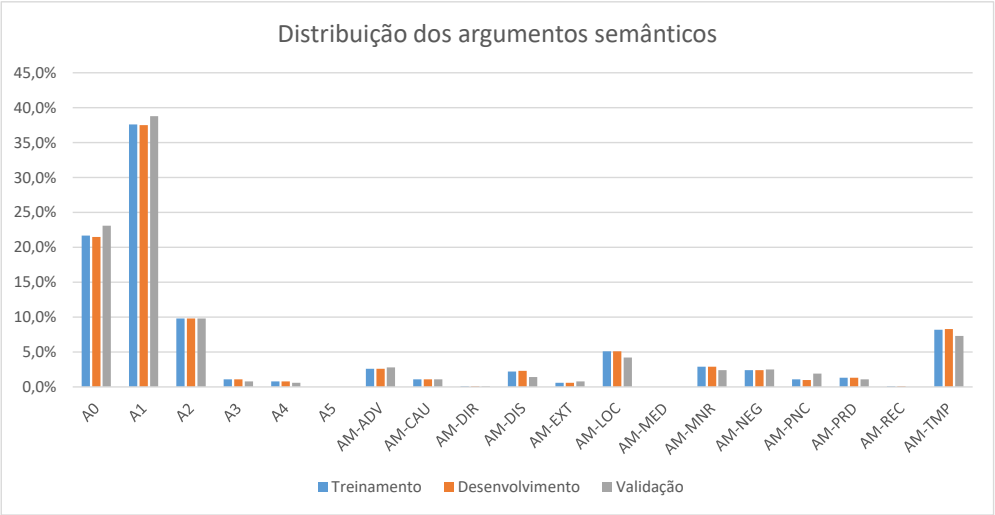

Figura 7: Gráfico de distribuição dos argumentos semânticos nos conjuntos de treinamento, desenvolvimento e validação.

4. Anotador de papéis semânticos supervisionado

Anotação de papeis semânticos (APS) é uma tarefa de aprendizado de máquina que estabelece uma relação de significado entre o verbo e outros componentes de uma dada sentença. Essa relação é definida através de rótulos associados aos trechos da sentença que foram identificados como argumentos do verbo analisado.

Esse capítulo é dedicado a detalhar o processo de modelagem da nossa abordagem para o problema da APS automática. Iniciamos pela simplificação da tarefa, que geralmente na literatura é composta de uma etapa de identificação de argumentos e uma segunda de classificação. Nosso classificador é treinado de forma a enxergar uma única tarefa de classificação, onde existe uma classe de não argumentos. A segunda diferença substancial está na forma de geração dos exemplos. Em nosso caso, todo token do conjunto de dados dá origem a um exemplo para o classificador, ao passo que na literatura geralmente essa construção é de constituinte a constituinte. Finalmente temos uma última diferença na classificação de argumentos, onde as etiquetas de papel semântico são atribuídas à raiz da subárvore de dependência do argumento ou invés de ser atribuída ao constituinte.

4.1. Geração de exemplos

Uma parte importante de qualquer sistema de aprendizado de máquina é a definição do como são gerados os exemplos encaminhados para o algoritmo de treinamento. A partir das observações contidas no *corpus* de entrada, é possível escolher diferentes abordagens para se agrupar e representar essas informações. Seria possível, por exemplo, que essas observações fossem agrupadas e moldadas na forma de grafos e que os exemplos fossem gerados de arco em arco. Essa decisão impacta na forma como os atributos são extraídos e por consequência na forma como é feita a classificação e pós-processamento destas.

Esse trabalho está mais interessado no uso das informações contextuais presentes na árvore sintática de dependência. Optou-se então pela geração de exemplos token a token, uma vez que qualquer nó dessa árvore, isto é, qualquer *token* da sentença, pode representar a eventual raiz de um argumento. Essa estratégia difere do que geralmente é visto na literatura, onde a opção é gerar exemplos de constituinte a constituinte através da árvore sintagmática. Além disso também optamos por não realizar nenhum tipo de poda nos exemplos, deixando a cargo do classificador a definição de quais candidatos realmente são argumentos.

4.1.1. Engenharia de atributos

Os atributos implementados nessa dissertação foram categorizados em 5 grupos diferentes. Cada um tenta capturar o contexto em torno do *token* de uma maneira diferente de forma a maximizar o potencial de generalização.

O primeiro grupo é formado de 5 atributos básicos do token que podem ser extraídos diretamente das colunas descritas na Tabela 2. Esses atributos estão listados na Tabela 8.

Atributo	Descrição
Form	Forma da palavra ou sinal de pontuação
Lemma	Lema *gold-standard* do *token*
POS	Etiqueta POS (classe gramatical) *gold-standard*
Morf	Atributos morfossintáticos *gold-standard*
Func	Função sintática do *token* para com seu regente na árvore de dependência

Tabela 8: Atributos básicos do token.

O segundo grupo tem como objetivo capturar o contexto do token na sentença através das informações dos seus vizinhos à esquerda e à direita. Aqui usamos uma janela de tamanho 7, o que significa dizer que são usadas informações dos 3 primeiros tokens à esquerda assim como dos 3 primeiros tokens à direita. A Figura 8 ilustra essa ideia.

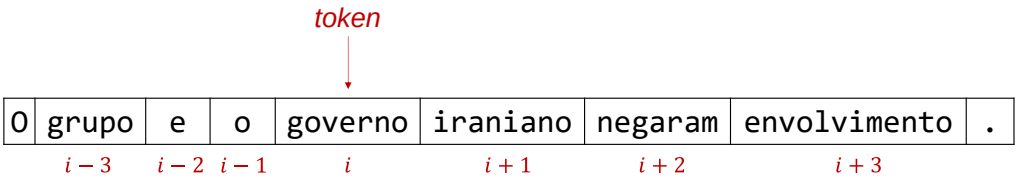

Figura 8: Exemplo de contexto do token usando uma janela de tamanho 7.

O único atributo desse grupo que não usa a ideia de janela é o *SentenceHash* por ser meramente um identificador. A Tabela 9 detalha os atributos que fazem parte dessa categoria.

Atributo	Descrição
LeftForm 1, 2 e 3	**Form** dos 3 *tokens* à esquerda
RightForm 1, 2 e 3	**Form** dos 3 *tokens* à direita
LeftFunc 1, 2 e 3	**Func** dos 3 *tokens* à esquerda
RightFunc 1, 2 e 3	**Func** dos 3 *tokens* à direita
LeftLemma 1, 2 e 3	**Lemma** dos 3 *tokens* à esquerda
RightLemma 1, 2 e 3	**Lemma** dos 3 *tokens* à direita
LeftPOS 1, 2 e 3	**POS** dos 3 *tokens* à esquerda
RightPOS 1, 2 e 3	**POS** dos 3 *tokens* à direita
SentenceHash	Identificador único da sentença. Se repete para proposições originadas da mesma sentença. Primeiro dos atributos inéditos mencionados em 7.2.

Tabela 9: Atributos de contexto na sentença.

A seguir é definido um grupo de atributos que tem a meta de expressar a relação existente entre o token, o predicado e a vizinhança em torno do predicado. Esses atributos são explicados na Tabela 10.

Atributo	Descrição
PredLemma	**Lemma** do verbo alvo
PredLeftLemma	**Lemma** do *token* à esquerda do verbo alvo
PredRightLemma	**Lemma** do *token* à direita do verbo alvo
PredPOS	**POS** do verbo alvo
PredLeftPOS	**POS** do *token* à esquerda do verbo alvo
PredRightPOS	**POS** do *token* à direita do verbo alvo

PredFunc	**Func** do verbo alvo
PredLeftFunc	**Func** do *token* à esquerda do verbo alvo
PredRightFunc	**Func** do *token* à direita do verbo alvo
PredicateDistance	Índice do verbo alvo subtraído do índice do *token* atual, podendo assumir valor negativo
PredMorf 1...*n*	Conjunto de 32 atributos morfossintáticos do verbo alvo, um para cada etiqueta listada na Tabela 4.
PassiveVoice	Indicador de voz passiva. Verdadeiro se o verbo alvo tiver **POS**=v-pcp e for precedido por *token* com **Lemma**=ser podendo ou não haver um *token* com **POS**=adv entre eles
PosRelVerb	Se o *token* está antes ou depois do verbo alvo

Tabela 10: Atributos de contexto com o predicativo alvo.

O atributo básico *Morf* é um campo multivalorado, ou seja, pode assumir simultaneamente diversas instâncias das 32 etiquetas definidas na Tabela 4. No trabalho de (ALVA-MANCHEGO, 2013) esse campo é representado como um único atributo categórico que, nos dados de treinamento, assume 110 valores distintos. Nesse trabalho, optamos por representar essas informações de forma independente usando 32 atributos categóricos diferentes, onde cada um deles pode assumir apenas os valores ausente e presente. Essa codificação foi usada apenas para o campo *Morf* do predicativo alvo.

Usando como exemplo a instância "PS|3S|IND", no primeiro modelo teríamos um mapeamento para um conjunto de 110 atributos binários, onde apenas um deles assumiria o valor 1. Já no segundo modelo "PS|3S|IND" seria separado em "PS", "3S" e "IND" gerando assim 3 atributos com valor 1 e 29 com valor 0. Mais detalhes sobre a binarização de atributos podem ser vistos na próxima seção. Essa representação é responsável por 32 dos atributos inéditos mencionados no capítulo 7.2. O atributo *PassiveVoice*, da forma como é definido aqui, também foi considerado inédito e contabilizado nas contribuições deste trabalho.

O próximo grupo trata dos atributos que contextualizam o token na árvore de constituintes. A maioria deles são atributos clássicos encontrados na literatura como *Path* (GILDEA e JURAFSKY, 2002) e *PartialPath* (PRADHAN, WARD e MARTIN, 2008). Alguns desses atributos fazem uso da propriedade núcleo do

sintagma. Para tanto elaboramos o Algoritmo 1 com base nas regras de identificação do núcleo dos sintagmas descritas em (ALVA-MANCHEGO, 2013).

Algoritmo 1: Identificação dos núcleos dos constituintes.

```
EncontrarNucleo
Entrada: c ← constituinte
Saida: w
  Se tipo de c = NP
    w ← primeiro filho de c com POS ∈ {substantivo, pronome}
  Se tipo de c = AP
    w ← primeiro filho de c com POS ∈ {adjetivo, determinante}
  Se tipo de c = ADVP
    w ← primeiro filho de c com POS ∈ {adverbio}
  Se tipo de c ∈ {VP, FCL, ICL}
    w ← primeiro filho de c com POS ∈ {verbo}
  Se tipo de c = PP
    w ← primeiro filho de c com POS ∈ {preposição}
  Se tipo de c = ADVP
    w ← primeiro filho de c com POS ∈ {adverbio}
  Senao
    w ← EncontrarNucleo do primeiro filho constituinte de c
```

A Tabela 11 descreve todos os atributos que estabelecem contexto na árvore sintagmática. Recomendamos a visualização simultânea da Figura 5 para melhor entendimento.

Atributo	Descrição
Head	**Forma** do núcleo do constituinte do qual o *token* faz parte
HeadFunc	**Func** do núcleo do constituinte do qual o *token* faz parte
HeadLemma	**Lemma** do núcleo do constituinte do qual o *token* faz parte
HeadPOS	**POS** do núcleo do constituinte do qual o *token* faz parte
Path	Caminho entre o constituinte do verbo alvo e o constituinte do *token* passando pelo menor ancestral comum entre ambos, conforme definido em (GILDEA e JURAFSKY, 2002)
PartialPath	Caminho entre o constituinte do *token* e o menor ancestral comum com o constituinte do verbo alvo, conforme definido em (PRADHAN, WARD e MARTIN, 2008)

PhraseType	Tipo de sintagma do qual o *token* faz parte
LeftPhraseType	Tipo de sintagma do primeiro irmão esquerdo do constituinte do *token*
RightPhraseType	Tipo de sintagma do primeiro irmão direito do constituinte do *token*
LeftPT 1, 2 e 3	Tipo de sintagma dos 3 *tokens* à esquerda
RightPT 1, 2 e 3	Tipo de sintagma dos 3 *tokens* à direita
DepthRelVerb	Profundidade do *token* subtraída da profundidade do verbo alvo

Tabela 11: Atributos de contexto na árvore de constituintes.

Por fim a última categoria de atributos descreve as relações do token na árvore de dependência. Até a presente data os autores desconhecem o uso destes em outros trabalhos relacionados, sendo esta, portanto, a primeira vez em que são definidos, contabilizando 23 dos atributos inéditos mencionados no capítulo 7.2. A Tabela 12 detalha esse conjunto. Para melhor entendimento recomendamos a visualização da Figura 9.

Atributo	**Descrição**
DepLemmaParent	**Lemma** do pai do *token*
DepLemmaGrandparent	**Lemma** do avô do *token*
DepLemmaChild 1, 2 e 3	**Lemma** dos 3 primeiros filhos do *token*
DepPOSParent	**POS** do pai do *token*
DepPOSGrandParent	**POS** do avô do *token*
DepPOSChild 1, 2 e 3	**POS** dos 3 primeiros filhos do *token*
DepPOSLeftSister 1, 2 e 3	**POS** dos 3 primeiros irmãos à esquerda do *token*
DepPOSRightSister 1, 2 e 3	**POS** dos 3 primeiros irmãos à direita do *token*
DepFuncGrandparent	**Func** do avô do *token*
DepFuncParent	**Func** do pai do *token*
DepFuncChild 1, 2 e 3	**Func** dos 3 primeiros filhos do *token*
DepPathFunc	Caminho de etiquetas **Func** entre o *token* e o verbo alvo passando pelo menor ancestral comum
DepPathPOS	Caminho de etiquetas **POS** entre o *token* e o verbo alvo passando pelo menor ancestral comum

Tabela 12: Atributos de contexto na árvore de dependência.

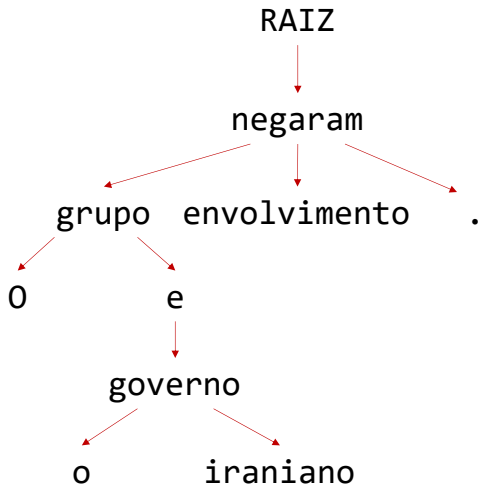

Figura 9: Exemplo de árvore de dependência em visualização vertical.

4.1.2. Binarização de atributos

Em sistemas de aprendizado de máquina dizemos que um exemplo x é representado pelo seu vetor de atributos $\vec{x}$ na forma $(x_1, x_2, \ldots, x_n)$. Os atributos descritos na seção anterior que definem esse conjunto estão representados na forma categórica, isto é, podem assumir um valor único a partir de uma lista pré-definida. Esse é o caso por exemplo do atributo **forma** que assume uma dentre as tantas palavras existentes no vocabulário do *corpus* PropBank.Br.

Embora essa representação seja bastante significativa para seres humanos, não é adequada para sistemas de base matemática e estatística, como é o caso do nosso classificador SVM. Se faz necessário então um processo de conversão desses dados categóricos em numéricos. Em especial no nosso caso cada atributo categórico é convertido em um vetor binário e esparso. A essa conversão damos o nome de *binarização*.

Logo, no conjunto de cores {vermelho, azul, verde}, azul poderia ser representado como {0, 1, 0}. Apesar de correta essa representação ainda não é a ideal, uma vez que cada instância de um atributo assumiria a forma de um vetor com *n* posições e *n-1* dessas posições estariam preenchidas com 0, sendo *n* o número de valores distintos do atributo. Ora, uma outra forma mais compacta de representar esse mesmo vetor é indicando qual a posição que está preenchida com

1. Nesse trabalho a binarização é atingida através do uso de um dicionário global que mantém um identificador numérico para cada par atributo-valor. A Figura 10 ilustra esse processo.

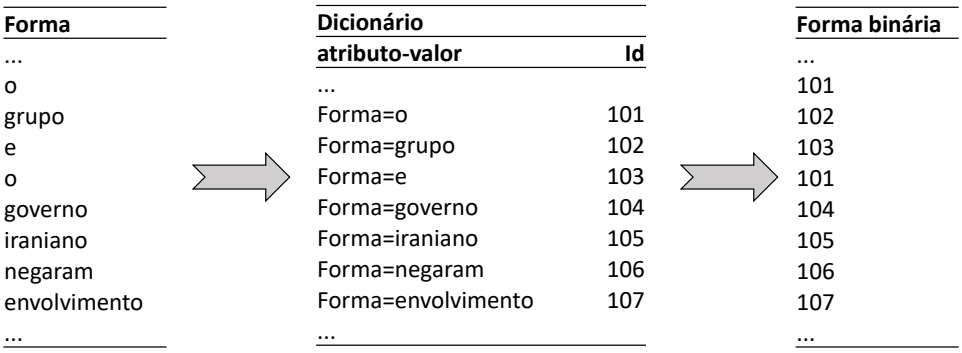

Figura 10: Exemplo de binarização do atributo forma.

4.2. Identificação e classificação de argumentos: um único classificador

Uma prática comum em sistemas de aprendizado de máquina supervisionado para APS é a divisão da tarefa de anotação em duas partes menores: identificação dos candidatos a argumentos do verbo e anotação dos argumentos pré-selecionados. Isso implica na implementação de um classificador para cada uma dessas tarefas, geralmente apresentando conjuntos de atributos diferentes. A literatura de PLN está repleta de estudos que usam essa mesma estratégia, como por exemplo a grande maioria, senão todos, os trabalhos listados no capítulo 2. Existem duas motivações para realizar a tarefa dessa forma: a primeira é que o uso de conjuntos de atributos diferentes pode melhorar o desempenho dos classificadores; a segunda é diminuição do tempo de treinamento no segundo classificador uma vez que a maioria dos exemplos será eliminada na primeira etapa.

No entanto essa técnica também tem suas desvantagens. A primeira desvantagem óbvia é a necessidade de se implementar e treinar dois classificadores diferentes. A segunda, não tão óbvia, é que o fato de usarmos dois classificadores também duplica todas as demais etapas de pesquisa e desenvolvimento que vem em seguida aumentando a complexidade do projeto como um todo. Esse é o caso por exemplo da regularização de domínio e da indução de atributos.

Outro fato relevante que contribui para a ideia de unificação é o recente desenvolvimento da ferramenta *liblinear*, que tem a capacidade reduzir drasticamente os tempos de treinamento em aplicações típicas de PLN além de oferecer um dos mais eficientes classificadores disponíveis atualmente (SVM).

Nesse trabalho optamos pela simplificação da tarefa em uma única etapa de classificação, onde existe uma classe de não argumentos. Usamos a ferramenta *liblinear* para reduzir com sucesso os tempos de treinamento de nosso modelo SVM e assim suprimimos a primeira motivação para o particionamento da tarefa. No capítulo 5 explicaremos como conseguimos minimizar nosso conjunto de atributos de forma a atingir a maior generalização possível do modelo e evitar o *overfitting*. Esse conjunto de atributos minimalista se mostra eficiente na anotação de papéis semânticos em uma única etapa atingindo desempenho equivalente ao estado-da-arte para APS automática em português. Dessa forma também suprimimos o segundo argumento a favor da divisão.

4.2.1. Parâmetros do classificador SVM

De forma a obter o maior desempenho possível de um classificador é necessário que usemos parâmetros adequados ao problema, tipo de modelo e a natureza dos dados. A implementação da *liblinear* que usamos durante o desenvolvimento deste trabalho, versão 2.1, oferece uma série de parâmetros que podem ser configurados para esse fim.

A primeira decisão que deve ser feita é qual implementação de modelo SVM que será usada no treinamento. São oferecidas 8 implementações diferentes para o problema multiclasse. A Tabela 13 e Figura 11 mostram um comparativo entre as 8 opções de *solver* disponíveis usando *10-fold cross validation* no conjunto de treinamento com os parâmetros padrão.

Implementação	Erro	Tempo
L2-regularized LR (primal)	2,80%	00:25:02
L2-regularized L2-loss SVC (dual)	2,84%	00:02:54
L2-regularized L2-loss SVC (primal)	2,82%	00:31:22
L2-regularized L1-loss SVC (dual)	2,88%	00:03:14
SVC by Crammer and Singer	2,95%	00:02:03
L1-regularized L2-loss SVC	2,67%	00:40:32

L1-regularized LR	2,73%	00:13:42
L2-regularized LR (dual)	2,78%	00:08:49

Tabela 13: Comparativo entre as 8 implementações de solver multiclasse da liblinear.

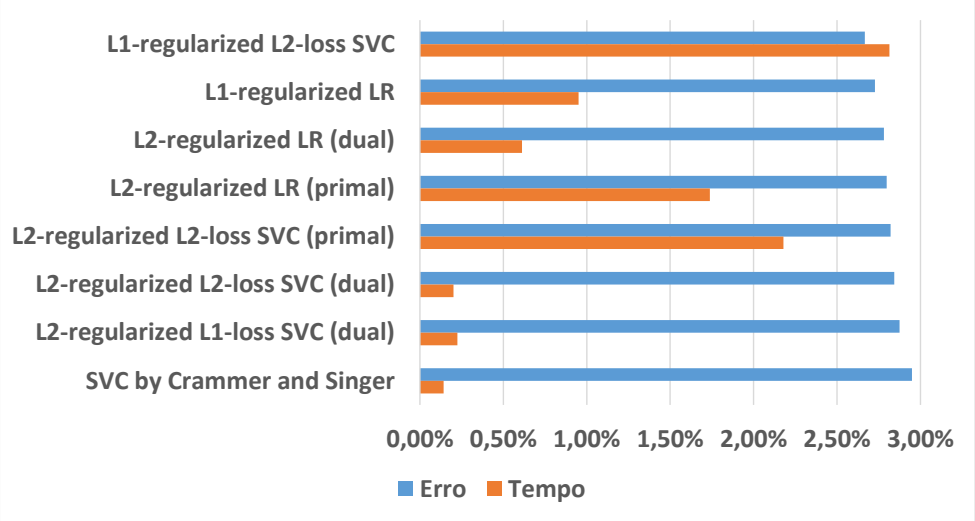

Figura 11: Gráfico comparativo do liblinear Implementação X Erro de treino + Tempo de execução.

Analisando o gráfico da Figura 11 podemos concluir que as diferenças no erro de treinamento não são muito significativas, todos os modelos ficam na faixa entre 2,67% e 2,95% de erro. Em contrapartida os tempos de execução mudam drasticamente tendo como pior resultado o modelo *L1-regularized L2-loss SVC* com 40m32s e o mais rápido sendo *SVC by Crammer and Singer* com o tempo de 2m3s. Como nossa abordagem depende tanto de desempenho quanto de velocidade optamos pelo classificador *L2-regularized L2-loss SVC (dual)*, por apresentar o melhor conjunto das características desejadas. O modelo obtém erro 0,11% menor que o *Crammer and Singer* e um tempo de execução apenas 51s mais lento. Esse não por acaso é o *solver* padrão da *liblinear* e recomendado pelos autores na maioria dos casos.

Uma vez definida a implementação a próxima etapa é determinar os valores dos parâmetros específicos dessa implementação. No caso desse trabalho precisaremos apenas definir qual será o valor do parâmetro C.

Conforme visto no capítulo 1.2, o objetivo de uma SVM é construir um modelo baseado nos exemplos de treinamento, tal que se consiga predizer a classe dos exemplos não vistos apenas olhando para seus atributos.

Dado um conjunto de treinamento contendo pares de atributos-classe $(\vec{x}_i, y_i), i = 1, \ldots, l \; onde \; \vec{x}_i \in \mathcal{R}^n \; e \; y_i \in \{1, -1\}^l$, uma SVM soluciona o seguinte problema de otimização:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i$$

$$\text{Com as restrições } \begin{cases} y_i(w^T \phi(\vec{x}_i) + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases}$$

A função real ϕ projeta os vetores $\vec{x}_i$ em um espaço de maior dimensionalidade para que o modelo SVM possa encontrar um hiperplano de margem máxima tornando o problema linearmente separável. $C > 0$ é o parâmetro de penalidade para o erro de classificação. Logo podemos dizer que o parâmetro C é responsável pelo equilíbrio entre o erro de classificação e o maximização da margem, onde um valor de C muito grande tende a causar *overfitting* e um valor de C muito pequeno tende a causar *underfitting*.

A ferramenta liblinear possui a funcionalidade de busca do parâmetro C através de uma heurística inteligente que usa validação cruzada para testar valores entre 2^{-20} ... 2^{10}. A implementação de nossa escolha não é suportada por essa funcionalidade, mas felizmente seu primo mais próximo, *L2-regularized L2-loss SVC (primal)*, é. Com essa funcionalidade, usando *10-fold cross-validation*, o classificador atingiu a maior acurácia de 97,27% usando valor de $C = 0{,}0625 \; ou \; 2^{-4}$.

4.3. Conclusões do capítulo

Nesse capítulo mostramos os detalhes de modelagem do nosso classificador, desde a geração dos exemplos até a parametrização da SVM usada para a construção do modelo. Os atributos usados são em grande parte os clássicos encontrados na literatura, mas também foi adicionado um grupo de atributos dedicados a representar as relações presentes na árvore de dependência.

Foi proposta a execução da tarefa em uma única etapa em oposição ao que geralmente é aceito como boa prática para a APS. Em grande parte, isso foi possível devido a implementação da ferramenta *liblinear* que gera modelos SVM com tempos de treinamento drasticamente mais baixos em aplicações típicas de PLN. O

segundo fator que possibilitou essa abordagem foi a minimização do conjunto de atributos através de uma técnica combinada de seleção que mostraremos no capítulo 5.

5. Regularização de domínio

Conforme descrito no capítulo 4 nossa estratégia de classificação supervisionada engloba o uso de uma grande quantidade de atributos propostos anteriormente na literatura de PLN além de incluir uns tantos outros criados para representar diversos cenários de contexto. Também vimos nos capítulos 1.3 e 1.4 que modelos onde a dimensão de atributos é tão grande ou maior que a dimensão de exemplos tendem a gerar *overfitting.* Nosso modelo após a etapa de binarização é composto de aproximadamente 300.000 atributos binários e 140.000 exemplos. As técnicas de seleção de atributos propostas nessa seção ajudam a compactar e simplificar o modelo de forma a maximizar a generalização e minimizar o *overfitting*.

5.1. Métricas de avaliação do modelo

Antes de prosseguir para as técnicas de seleção propriamente ditas é necessário introduzir como é feita a comparação entre os modelos, de forma a saber qual é realmente o melhor. Até o momento trabalhamos apenas com a medida de acurácia da validação cruzada. Apesar de significativa essa medida sozinha não é o bastante para determinar a qualidade do modelo. Isso se deve ao fato da maioria dos exemplos do nosso conjunto de dados ser pertencente a classe de não-argumento. Conforme as informações da Tabela 7 podemos concluir que nosso conjunto de treinamento possuí 122.565 exemplos da classe nula e apenas 13.495 exemplos que efetivamente são argumentos. Ou seja, um classificador que atribua nulo para todos os exemplos terá cerca de 90% de acurácia.

Existem formas mais abrangentes de medir a qualidade de um modelo, 3 delas são clássicas de problemas de PLN e são exatamente as usadas nas *Shared Tasks da CoNLL*. Estamos falando de *precisão, cobertura e medida F* (conhecida também como F1).

Precisão é uma medida de exatidão que mede a fração de elementos anotados que são relevantes. Ela é definida pela fórmula:

$$P = \frac{p}{p + fp}$$

$$onde \begin{cases} p = \ número\ de\ anotações\ positivas \\ fp = número\ de\ anotações\ falso - positivas \end{cases}$$

Cobertura, também conhecida como sensibilidade, mede a fração de elementos relevantes que foi anotada. Sua fórmula é:

$$C = \frac{p}{p + fn}$$

$$onde \begin{cases} p = \ número\ de\ anotações\ positivas \\ fn = número\ de\ anotações\ falso - negativas \end{cases}$$

A saber: positivas são as anotações corretas feitas pelo classificador, falso-positivas são as anotações incorretas e falso-negativas são as anotações que o classificador deixou de realizar. A Figura 12 mostra ambas as formulas de uma forma mais intuitiva.

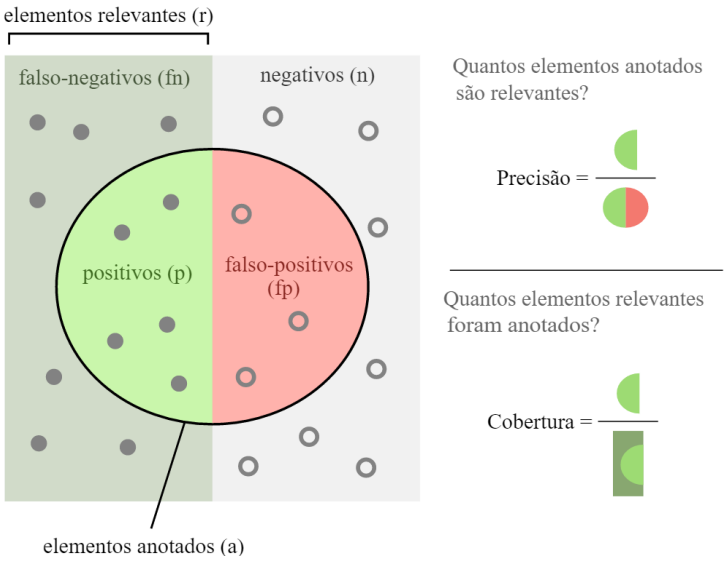

Figura 12: Precisão e Cobertura. Traduzida da Wikipedia.

Medida F (F1) combina precisão e cobertura usando uma média harmônica. Sua fórmula é definida como:

$$F1 = 2 \times \frac{P \times C}{P + C}$$

$$onde \begin{cases} P = Precisão \\ C = Cobertura \end{cases}$$

Aplicando as fórmulas no classificador que atribui nulo para todas as etiquetas teríamos 0 para precisão, cobertura e F1. Um exemplo melhor para ilustrar o uso das fórmulas seria um classificador que atribuísse com 100% de acerto a classe A1 e anulasse as demais. Nesse cenário teríamos:

$$Acurácia = \frac{5.077 + 122.565}{136.036} = 0{,}938 = 93{,}8\%$$

$$Precisão = \frac{5.077}{5.077 + 0} = 1 = 100\%$$

$$Cobertura = \frac{5.077}{5.077 + 8.418} = 0{,}376 = 37{,}6\%$$

$$F1 = 2 \times \frac{1 \times 0{,}376}{1 + 0{,}376} = 0{,}546 = 54{,}6\%$$

5.2. Seleção binária

Dado um dataset binário $D = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)\}$ o objetivo de um perceptron multiclasse é construir um modelo $W = \{\vec{w}_1, \vec{w}_2, \dots, \vec{w}_m\}$ tal que $\hat{y} = argmax_y f(\vec{x}, y) \cdot \vec{w}$, onde n é o número de exemplos, m o número de atributos, o vetor $\vec{x}$ possui m elementos e o vetor $\vec{w}$ possui tantos elementos quanto forem o número de classes.

Em (MOTTA, 2014) é proposta uma estratégia para seleção de atributos binários baseada no conceito de que atributos mais relevantes estão envolvidos em um maior número de atualizações de sinapse. Para tanto é treinado um perceptron esparso multiclasse modificado para guardar um registro $R = \{r_1, r_2, \dots, r_m\}$, onde r_i é o número de atualizações de $\vec{w}_i$ e i é o índice do atributo. Dessa forma basta aplicar um filtro qualquer a R para realizar algum tipo de seleção, como por exemplo a seleção de todos os atributos com r acima de um limiar l.

Nossa abordagem para a seleção de atributos binários consiste em adaptar o esse conceito para o uso com classificadores SVM. A ferramenta liblinear não dispõe de uma funcionalidade que grave um registro de atualizações de $\vec{w}$. No entanto o modelo gerado pela implementação *L2-regularized L2-loss SVC (dual)* segue a estratégia multiclasse *one-versus-all*, o que significa dizer que o modelo

segue o formato $W = \{\vec{w}_1, \vec{w}_2, \dots, \vec{w}_m\}$. Sabendo que $\vec{w}_i$ representa a influência do atributo i em relação a cada um dos possíveis valores de y, e que um elemento $w_{i,j}$ pode assumir valores positivos ou negativos, propomos que a relevância do atributo possa ser calculada como $max(|\vec{w}_i|)$. Sendo assim é possível aplicar filtros similares aos definidos no IFIS. Para ter controle sobre o tamanho do modelo resultante filtramos os atributos por quantidade. Esse conceito é expressado no Algoritmo 2.

Algoritmo 2: Seleção de atributos binários baseada no IFIS.

```
SelecionarAtributosBinarios
Entrada: D = {(x⃗_1, y_1), (x⃗_2, y_2), ..., (x⃗_n, y_n)} Dataset binário,
         A número de atributos
Saida: S
  M ← ∅
  S ← ∅
  W ← Treinar(D)
  Para cada w⃗_i ∈ W
    M ← M ∪ {(max(|w⃗_i|), i)}
  M ← OrdenarDescendente(M)
  Para j = 0 até A
    (max, id) ← m_j ∈ M
    S ← S ∪ {(id)}
```

Para escolher a força de regularização usada na seleção combinada definimos 10 faixas com valores entre 70% e 99% de redução nos atributos binários. Sabendo que essa nova composição de atributos certamente afetaria a eficiência do parâmetro C do modelo SVM, fizemos uma nova medição para cada uma das faixas estipuladas. Em seguida aferimos o resultado de cada um dos modelos usando *10-fold cross validation* no conjunto de treinamento assim como medida F no conjunto de validação. O primeiro treinamento com o conjunto completo foi feito com *C* = 0,0625 e o treinamento do conjunto regularizado foi feio de acordo com o novo *C* medido para cada uma das faixas. Os resultados são expostos na Tabela 14 e Figura 13.

Regularização	C	F1 validação	*10-fold*
99,0%	0,125	78,38%	97,63%
98,0%	0,125	78,78%	97,70%
97,0%	0,125	78,71%	97,69%

96,0%	0,125	79,02%	97,65%
95,0%	0,25	78,25%	97,62%
90,0%	0,25	78,24%	97,43%
85,0%	0,25	78,42%	97,31%
80,0%	0,25	78,08%	97,25%
75,0%	0,125	78,63%	97,27%
70,0%	0,0625	78,91%	97,27%

Tabela 14: Resultados de classificação por faixa de regularização binária.

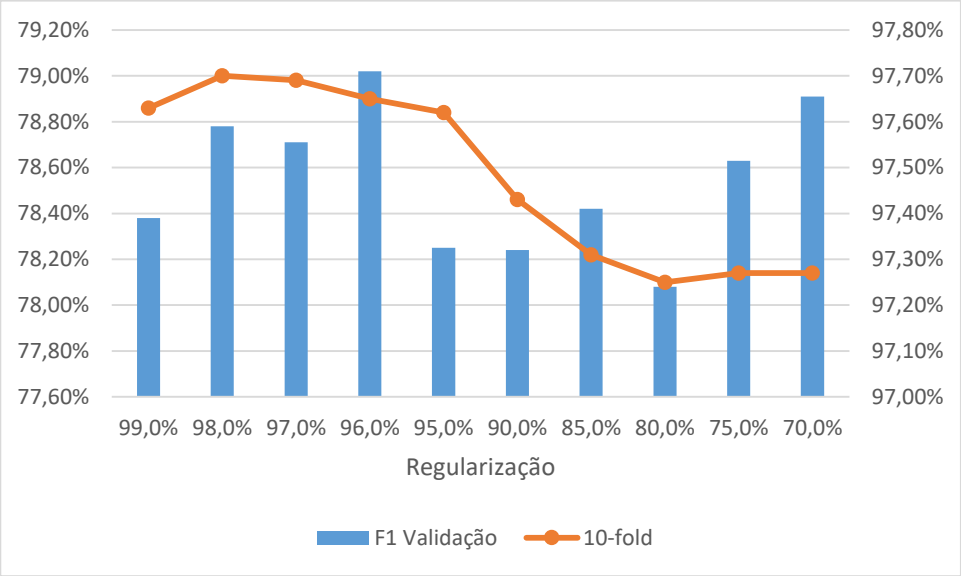

Figura 13: Gráfico comparativo das faixas regularização binária.

A análise do gráfico nos permite rapidamente concluir que a regularização de maior poder de generalização é a de 96%. Um outro fator interessante é que essa faixa de redução além de compactar e simplificar o modelo acarretou em um pequeno ganho tanto em medida F como em acurácia de validação cruzada uma vem que o conjunto de validação com todos os atributos binários obtém 97,27% e 78,88% respectivamente.

5.3. Seleção baseada em *greed hill climbing*

A próxima técnica de seleção de atributos é categórica, ou seja, ela tem o objetivo de restringir categorias inteiras de atributos ao invés de valores específicos dos mesmos. Para exemplificar e contrastar as diferentes abordagens podemos supor que enquanto a seleção binária seleciona valores específicos do atributo ***lemma*** como {ser, estar, fazer} a seleção categórica seleciona o atributo ***lemma*** como um todo mantendo todos os seus valores.

Nossa estratégia de seleção categórica é baseada em *greed hill climbing*, que tem como objetivo ampliar o conjunto de atributos selecionados de um em um, desde que esse novo atributo aumente o desempenho do modelo anterior. O algoritmo para quando não houverem mais atributos a adicionar. A diferença do algoritmo tradicional para a nossa abordagem é que nós inserimos uma segunda etapa, onde o modelo será reduzido desde que o atributo retirado não diminua o desempenho do modelo. Fazemos isso até não restarem atributos a serem removidos. A teoria por trás desse procedimento é que após a inclusão de *n* atributos no modelo alguns atributos adicionados nos ciclos anteriores podem ter ficado redundantes. O Algoritmo 3, Algoritmo 4 e Algoritmo 5 detalham esse processo.

Algoritmo 3: Seleção de atributos categóricos baseada em greed hill climbing.

```
SelecionarAtributosCategoricos
Entrada: D = {(x⃗_1, y_1), (x⃗_2, y_2), ..., (x⃗_n, y_n)} Dataset categórico,
         C = {c_1, c_2, ..., c_m} Lista de atributos categóricos
Saida: S
  S ← ∅
  C̄ ← ∅
  Para cada c_i ∈ C
    D̄ ← Binarizar(D, {c_i})
    v ← AvaliacaoCruzada(D̄, 10)
    C̄ ← C̄ ∪ {(v, c_i)}
  C̄ ← OrdenarDescendente(C̄)
  v ← 0
  Faça
    (S, v) ← IncrementarSelecao(D, C̄, S, v)
    (S, v) ← DecrementarSelecao(D, C̄, S, v)
  Enquanto S atualizar
```

Algoritmo 4: Adiciona atributos a uma seleção anterior.

```
IncrementarSelecao
Entrada: D = {(x⃗_1, y_1), (x⃗_2, y_2), ..., (x⃗_n, y_n)} Dataset categórico,
         C = {c_1, c_2, ..., c_m} Lista de atributos categóricos
         S = {s_1, s_2, ..., s_p} Lista de atributos selecionados
         v Valor da avaliação cruzada de S
Saida: S,
       v
  C ← C − S
  Para cada c_i ∈ C
    S̄ ← S ∪ {(c_i)}
    D̄ ← Binarizar(D, S̄)
    v̄ ← AvaliacaoCruzada(D̄, 10)
    Se v̄ > v
      S ← S̄
      v ← v̄
  Retornar(S, v)
```

Algoritmo 5: Remove atributos desnecessários de uma seleção anterior.

```
DecrementarSelecao
Entrada: D = {(x⃗_1, y_1), (x⃗_2, y_2), ..., (x⃗_n, y_n)} Dataset categórico,
         C = {c_1, c_2, ..., c_m} Lista de atributos categóricos
         S = {s_1, s_2, ..., s_p} Lista de atributos selecionados
         v Valor da avaliação cruzada de S
Saida: S,
       v
  C ← C ∩ S
  Para cada c_i ∈ C
    S̄ ← S − {(c_i)}
    D̄ ← Binarizar(D, S̄)
    v̄ ← AvaliacaoCruzada(D̄, 10)
    Se v̄ ≥ v
      S ← S̄
      v ← v̄
  Retornar(S, v)
```

Após a execução do algoritmo acima o modelo resultante mantém apenas 49 dos 114 atributos categóricos e 201.000 dos 300.000 atributos binários, significando uma redução de 57% e 33% respectivamente. A Tabela 15 mostra uma lista dos atributos mantidos.

Básicos	Contexto da sentença	Contexto do predicado
Feat	RightForm 1	Predicate
Func	RightFunc 1	PredPOS
POS	LeftLemma 1 e 2	PredLeftPOS
Lemma	RightLemma 1 e 2	PredRightPOS
	RightPOS 1 e 3	PredRightFunc
	PassiveVoice	PredicateDistance
	PosRelVerb	PredMorf 1P, 3S, FUT, M, PR, S, SUBJ

	Contexto de constituintes	Contexto de dependência
	Head	DepLemmaParent
	HeadFunc	DepLemmaChild 1, 2 e 3
	HeadLemma	DepPOSParent
	HeadPOS	DepPOSChild 1 e 3
	Path	DepPOSLeftSister 1
	PartialPath	DepPOSRightSister 1
	PhraseType	DepFuncParent
	LeftPhraseType	DepFuncChild 1 e 2
		DepPathFunc
		DepPathPOS

Tabela 15: Atributos mantidos após seleção categórica.

A seguir treinamos o modelo novamente usando apenas os atributos selecionados e medimos o resultado no conjunto de validação. Os resultados mostram um incremento na medida F de quase 1% como podemos ver na Tabela 16.

Atributos	F1 validação	*10-fold*
Todos	78,88%	97,27%
Seleção categórica	79,78%	97,45%

Tabela 16: Desempenho comparativo no conjunto de validação antes e depois da seleção categórica.

5.4. Seleção composta

O último passo da nossa abordagem é combinar as seleções categórica e binaria em um único modelo. Para tanto realizamos primeiro o treinamento com os atributos pré-selecionados pela regularização categórica, mostrados na Tabela 15, usando C=0,0625. O modelo resultante é filtrado novamente, dessa vez usando a seleção binária, conforme procedimentos do Algoritmo 2. O parâmetro A é regulado para 96% da quantidade de atributos binários do modelo original, ou seja, 11.120. Finalmente realizamos um último treinamento apenas com esses atributos usando C=0,125. Demostramos os resultados obtidos com o conjunto de validação na Tabela 17 e Figura 14.

Atributos	C_1	C_2	F1 validação	*10-fold*
Todos	0,0625	-	78,88%	97,27%
Seleção binária	0,0625	0,125	79,02%	97,65%
Seleção categórica	0,0625	-	79,78%	97,45%
Seleção composta	0,0625	0,125	79,58%	97,60%

Tabela 17: Comparativo entre as seleções binária, categórica e composta no conjunto de validação.

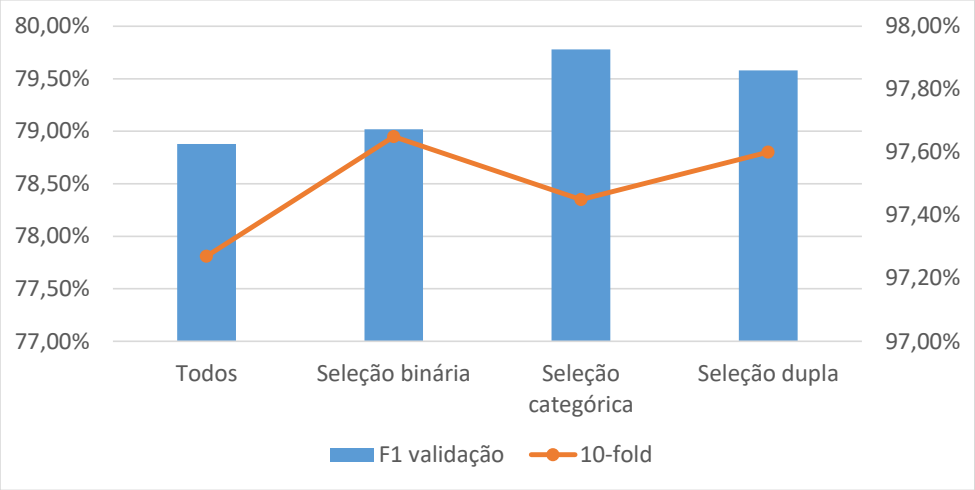

Figura 14: Gráfico comparativo das diferentes seleções aplicadas ao conjunto de validação.

Apesar do gráfico mostrar uma pequena queda de medida F quando aplicamos a regularização binária sobre a categórica podemos ver que o oposto ocorreu com a medição aferida pela validação cruzada. Isso nos leva a crer que de fato houve uma maior generalização, e que essa queda de 0,2% na medida F pode ser desprezada em favor de um modelo muito mais compacto (cerca de 94% menor). Também podemos concluir a regularização composta é mais eficiente que a binária isolada, já que houve um aumento de mais de 0,5% em medida F e um redução de apenas 0,05% na medição da validação cruzada. Todos esses fatos nos levam a crer que o modelo gerado pela seleção composta é o mais adequado para compactar e generalizar o modelo simultaneamente.

5.5. Conclusões do capítulo

Nesse capítulo exploramos as duas técnicas de regularização propostas nesse trabalho. Ambas as seleções têm o objetivo de minimizar a quantidade de atributos no modelo (quase) sem perda de desempenho. No entanto a seleção categórica atua em conjuntos de atributos, como por exemplo selecionando o atributo ***lemma***, ao passo que a binária seleciona valores discretos desses atributos, como {ser, estar, fazer}.

Mostramos que todas as técnicas têm o potencial de generalizar o modelo, inclusive aumentando o desempenho em casos onde existem muitos atributos irrelevantes ou redundantes. Finalmente analisamos os resultados isolados e combinados dessas abordagens onde concluímos que o uso da regularização composta permite uma maior generalização do modelo.

6. Experimentos e Resultados

Essa seção tem o intuito de detalhar os resultados de nossos experimentos usando arquitetura que foi demonstrada nos capítulos anteriores. De forma a possibilitar a comparação empírica desses experimentos com outros trabalhos usamos a padronização criada pela competição internacional *CoNLL 2005 Shared Task*. Nela são estabelecidas a formalização da tarefa, os conjuntos de dados e as métricas de avaliação fazendo com que sistemas de APS automática possuam configurações de entrada e saída comuns e comparáveis.

Como conjunto de dados usamos o *corpus* PropBank.Br 1.1 que, conforme descrito no capítulo 3, possui características compatíveis com as definições da competição. As métricas de avaliação, descritas no capítulo 5.1, são calculadas diretamente pelo *script* oficial da competição (*srl-eval.pl*[15]) gerando os placares de precisão, cobertura e medida F (F1).

Assim, como estabelecido no capítulo 5, treinamos primeiramente o classificador com o conjunto de atributos da Tabela 15 usando os dados de treinamento e C=0,0625. É importante lembrar que nessa etapa estamos usando o conjunto completo de treinamento sem a partição de 10% para validação que foi usada nos capítulos anteriores. Em seguida realizamos o procedimento de seleção binária nesse modelo para remover 96% dos atributos e treinamos novamente com C=0,125.

A Tabela 18 detalha os resultados obtidos pelo nosso classificador sobre o conjunto de testes após passar pelas duas etapas de regularização. Apenas como referência incluímos também os resultados do classificador sem nenhuma regularização e somente com a regularização binária. É importante ressaltar que o número de atributos binários é cerca de 8% maior no conjunto de treinamento completo e por isso temos um aumento de 11.120 para 11.960 no conjunto regularizado final.

[15] http://www.cs.upc.edu/~srlconll/srl-eval.pl

Regularização	C_1	C_2	Precisão	Cobertura	F1	Atributos
Nenhuma	0,0625	-	80,27%	80,82%	80,55%	299.000
Seleção binária	0,0625	0,125	80,14%	80,82%	80,48%	11.960
Seleção categórica	0,0625	-	81,12%	81,68%	81,40%	201.000
Seleção composta	0,0625	0,125	82,17%	82,88%	82,52%	11.960

Tabela 18: Resultados de cada etapa de regularização no conjunto de testes.

O gráfico da Figura 15 permite a visualização das mesmas informações de uma maneira mais objetiva. Conforme podemos constatar facilmente a regularização composta aumentou o desempenho no conjunto de testes em quase 2% de medida F ao mesmo tempo em que tornou o modelo mais compacto e genérico. Também é possível notar uma melhora de desempenho ao usar a seleção binária sobreposta a seleção categórica confirmando a hipótese de que a composição generaliza melhor o modelo.

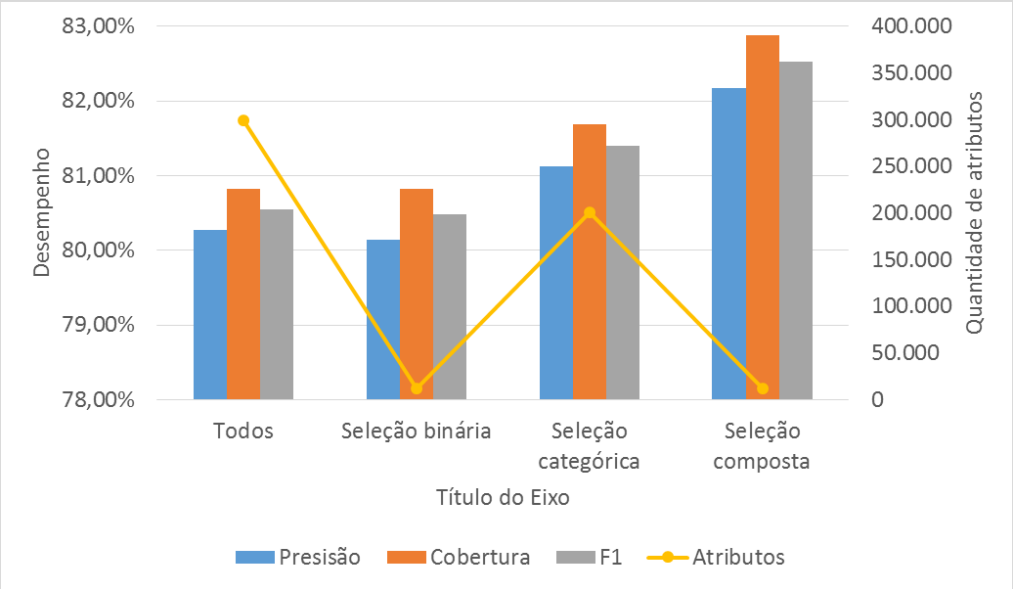

Figura 15: Gráfico comparativo entre as etapas de regularização nos dados de teste.

A seguir apresentamos na Tabela 19 os resultados individuais alcançados pelo sistema em cada uma das 16 etiquetas de papel semântico disponíveis no conjunto de testes.

Etiqueta	Corretas	Excedentes	Faltantes	Precisão	Cobertura	F1
Global	484	105	100	82,17%	82,88%	82,52%
A0	111	10	13	91,74%	89,52%	90,61%
A1	204	24	30	89,47%	87,18%	88,31%
A2	52	24	12	68,42%	81,25%	74,29%
A3	1	0	1	100,00%	50,00%	66,67%
A4	2	0	3	100,00%	40,00%	57,14%
AM-ADV	11	2	9	84,62%	55,00%	66,67%
AM-CAU	2	2	0	50,00%	100,00%	66,67%
AM-DIR	0	0	1	0,00%	0,00%	0,00%
AM-DIS	8	5	2	61,54%	80,00%	69,57%
AM-EXT	0	0	1	0,00%	0,00%	0,00%
AM-LOC	23	5	6	82,14%	79,31%	80,70%
AM-MNR	7	9	11	43,75%	38,89%	41,18%
AM-NEG	21	1	0	95,45%	100,00%	97,67%
AM-PNC	2	2	2	50,00%	50,00%	50,00%
AM-PRD	2	4	3	33,33%	40,00%	36,36%
AM-TMP	38	17	6	69,09%	86,36%	76,77%

Tabela 19: Resultados finais do sistema detalhados por rótulo nos dados de teste.

Conforme o esperado o sistema obteve um desempenho melhor nas etiquetas que eram menos esparsas no conjunto de treinamento, como A0 e A1. A distribuição dessas etiquetas foi apresentada no capítulo 3.4 através da Figura 7 e da Tabela 7. Um fato interessante é que o sistema foi capaz de generalizar melhor as etiquetas AM-NEG, que contém apenas 2,4% das instâncias de treinamento, e AM-LOC (5,1%) do que A2 (9,8%) e AM-TMP (8,2%), evidenciando que alguns rótulos de APS são mais fáceis de aprender do que outros. Todas as demais etiquetas possuem representatividade inferior a 3% nos dados de treinamento, sendo assim difíceis de serem aprendidas adequadamente. As variações no desempenho são decorrentes da presença de exemplos mais fáceis ou mais difíceis nos dados de teste.

Finalmente apresentamos na Tabela 20 os resultados do nosso classificador comparados aos melhores sistemas de APS automática para português relatados até o momento. O *Baseline* que apresentamos aqui é um classificador simples baseado

em regras para predizer apenas as etiquetas A0, A1 e AM-NEG proposto por (ALVA-MANCHEGO, 2013).

Sistema	Ano	Algoritmo	Precisão	Cobertura	F1
Este Trabalho	**2016**	**SVM**	**82,17%**	**82,88%**	**82,52%**
Alva-Manchego	2013	Máxima Entropia	83,00%	81,70%	82,30%
Fonseca	2013	Rede Neural	68,97%	67,04%	67,99%
Baseline	2013	Regras fixas	64,60%	40,90%	50,10%

Tabela 20: Resultados finais comparados com os melhores sistemas de APS para português.

Como podemos ver o classificador proposto neste trabalho atinge desempenho ligeiramente superior ao atual estado da arte mesmo realizando a tarefa de APS automática simplificada em uma única etapa. Em grande parte, isso se deve a estratégia composta de regularização que foi capaz de reduzir drasticamente o tamanho do modelo ao mesmo tempo em que elevou o desempenho. Vimos também que a seleção categórica sozinha não é o suficiente para atingir a melhor generalização do modelo devido ao fato de ainda possuir muitos atributos redundantes e irrelevantes em nível binário.

Finalmente concluímos que a combinação de diferentes técnicas de regularização pode contribuir bastante para a generalização e compactação do modelo, permitindo assim economizar recursos preciosos como tempo de treinamento, memória e processamento além de aumentar o desempenho do classificador nos casos onde existem muitos atributos irrelevantes ou redundantes.

6.1. Conclusões do capítulo

Nesse capítulo mostramos os resultados empíricos do sistema proposto e os comparamos com os melhores sistemas de APS automática para português relatados até a presente data. Constatamos que a técnica composta de regularização de domínio é bastante eficiente quando o modelo possui muitos atributos irrelevantes ou redundantes. Essa técnica se mostrou capaz de superar o atual estado da arte mesmo que por uma pequena margem.

7. Conclusões

7.1. Resumo do trabalho

Nessa dissertação, desenvolvemos um sistema APS para a língua portuguesa usando modelagem SVM através da ferramenta *liblinear*. Além disso desenvolvemos 57 atributos categóricos que foram agregados a outros 57 na construção do modelo na tentativa de uma melhor descrição das relações intrínsecas a tarefa. O conjunto total de atributos binários resultante dessa engenharia de atributos possui cerca de 300 mil atributos em contraste com o número de exemplos que gira em torno de 130 mil.

É de conhecimento geral que modelos muito complexos, onde o número de parâmetros é maior que o número de observações, correm grande risco de incorrer em *overfitting*. De forma a maximizar a generalização do modelo e evitar esse fenômeno propusemos a regularização de domínio em duas etapas combinadas. O modelo resultante dessa estratégia é extremamente compacto e simples, mantendo apenas 4% dos atributos binários originais ao mesmo tempo em que eleva o desempenho do classificador.

Contrário ao que é geralmente encontrado na literatura esse trabalho simplifica a tarefa de APS reduzindo as fases de identificação e classificação de argumentos a um único classificador SVM ao mesmo tempo em que elimina uma etapa de pré-processamento (poda). Isso foi possível graças a velocidades de treinamento inferiores a 25 segundos possibilitadas pelo uso da ferramenta escolhida.

O sistema desenvolvido por essa dissertação está avaliado com o script oficial da tarefa compartilhada de APS da *CoNLL-2005* obtendo um desempenho de aproximadamente 82,17% de precisão, 82,88% de cobertura e 82,52% de F1 na classificação sendo assim ligeiramente melhor (em medida F) que o atual estado-da-arte (ALVA-MANCHEGO, 2013) que detém como resultados 83,0% de precisão, 81,7% de cobertura e 82,3% de F1.

Apesar do sistema apresentado neste trabalho superar o estado-da-arte para classificação APS em português ainda estamos cerca de 5 pontos percentuais abaixo do estado-da-arte para o idioma inglês americano[16] (ROTH e WOODSEND, 2014). É sabido que os recursos lexicais para o idioma inglês são bem mais abundantes e atribuímos esse melhor desempenho principalmente a diferença de tamanho entre o corpus americano e o corpus brasileiro (o americano possui cerca de dez vezes mais sentenças). Desta forma é notório que no português ainda temos muito a avançar na criação de recursos lexicais, o que deve ajudar enormemente no desempenho de aplicações de PLN baseadas em aprendizado de máquina.

7.2. Contribuições

A simplificação da tarefa de APS em uma única etapa, ao invés de separa-la em uma fase de identificação de atributos e uma segunda de classificação. Além disso eliminamos a poda como etapa de pré-processamento.

Desenvolveu-se um conjunto de 57 novos atributos (dos 114 usados nesse modelo) que usam diversas informações baseadas nas árvores sintagmáticas e de dependência. Vários desses em trabalho conjunto com um especialista em linguística.

Implementou-se um algoritmo baseado na técnica *greedy hill climbing* para realizar a seleção de atributos categóricos.

Propôs-se uma metodologia de seleção de atributos inspirada em (MOTTA, 2014), porém voltada para modelos SVM, reduzindo a quantidade de atributos binários usados no modelo em 96%.

A combinação dessas duas estratégias de regularização compactou o número de atributos binários para apenas 4% dos originais ao mesmo tempo em que elevou o desempenho em quase 2% de medida F.

Por fim, criou-se um classificador de APS para o idioma português que usa o corpus PropBank.br e atinge desempenho ligeiramente superior ao estado-da-arte. A avaliação é feita pelo script oficial da *CoNLL 2005 Shared Task* obtendo 82,17% de precisão, 82,88% de cobertura e 82,52% de F1.

[16] https://github.com/microth/mateplus

7.3. Trabalhos futuros

Como trabalhos futuros sugerimos abordar 4 pontos, discutidos a seguir.

Explorar outras técnicas de regularização de domínio e indução de atributos. O uso de técnicas diferentes das abordadas neste trabalho pode influenciar no desempenho e na generalização do modelo como um todo.

A investigação de expressões multi-vocabulares (MWE) pode trazer um ganho de desempenho numa aplicação do mundo real. Durante esse trabalho foram feitos experimentos promissores com as MWE, porém esses não se reverteram em um aumento de desempenho no sistema proposto. Isso se deve ao fato de o Propbank.br já possuir as MWE mais frequentes unificadas em um único elemento lexical.

Incorporar dados de outras fontes de informações semânticas como a VerbNet.Br e a WordNet.Br. O uso desses recursos pode ser aproveitado para agregar conhecimento linguístico ao sistema possivelmente elevando o desempenho da classificação. Uma forma simples seria a criação de novos atributos que codificassem algumas dessas características.

Acrescentar ou substituir o corpus Propbank.br por outros com anotações semânticas. No decorrer desse projeto usamos como base de treino e classificação o corpus Propbank.br 1.1. Uma forma de afetar diretamente os resultados do sistema é substituir o corpus ou unifica-lo com outras fontes de maneira a aumentar o número de exemplos para o aprendizado. Duas possibilidades nesse contexto são o Propbank.br 2.0 e uma possível versão semântica do Google universal. Publicado recentemente o Propbank.br 2.0 contém 8.350 instâncias do gênero jornalístico anotadas com papéis semânticos, mas em oposto a versão 1.1 não possui as árvores sintáticas revisadas por seres humanos. Já o Google Universal desperta interesse, mas ainda não possui uma versão com anotação de papéis semânticos, o que pode vir a ocorrer em breve.

8. Referências Bibliográficas

AFONSO, S. et al. **Floresta sintá(c)tica:** a treebank for Portuguese. Proceedings of LREC 2002. Las Palmas: ERLA. 2002.

ALVA-MANCHEGO, F. E. **Anotação automática semissupervisionada de papéis semânticos para o português do Brasil.** Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação. São Carlos. 2013.

AMANCIO, M. A.; DURAN, M. S.; ALUISIO, S. M. Automatic question categorization: a new approach for text elaboration. **Procesamiento del lenguaje natural**, v. 46, p. 43-50, 2010.

BABKO-MALAYA, O. **PropBank Annotation Guidelines**. University of Colorado. Boulder. 2005.

BAKER, C. F.; FILLMORE, C. J.; LOWE, J. B. The berkeley framenet project. **Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics**, Stroudsburg, USA, v. 1, p. 86-90, 1998.

BERMINGHAM, M. L. et al. Application of high-dimensional feature selection: evaluation for genomic prediction in man. **Scientific Reports**, v. 5, 2015.

BICK, E. **The Parsing System Palavras - Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Famework**. Department of Linguistics, University of Aarhus. Aarhus, Denmark. 2002.

BICK, E. **Noun Sense Tagging:** Semantic Prototype Annotation of a Portuguese. Proceedings of TLT 2006. Prague, Czech: Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University. 2006. p. 127-138.

BICK, E. **Automatic semantic role annotation for portuguese.** Proceedings of TIL 2007 - 5th Workshop on Information and Human Language Technology. Rio de Janeiro: Anais do XXVII Congresso da SBC. 2007. p. 1713-1716.

BLAKE, B. J. **Case - Cambridge Textbooks in Linguistics**. 2ª. ed. Victoria: Cambridge University Press, 2001. ISBN 9780521014915.

BUCHHOLZ, S.; MARSI, E. **CoNLL-X shared task on multilingual dependency parsing**. Proceedings of the Tenth Conference on Computational Natural Language Learning. New York: Association for Computational Linguistics. 2006. p. 149-164.

CARRERAS, X.; MÁRQUEZ, L. **Introduction to the CoNLL-2004 Shared Task:** Semantic Role Labeling. Proceedings of CoNLL-2004. Boston: ACL. 2004. p. 89-97.

CARRERAS, X.; MÁRQUEZ, L. **Introduction to the CoNLL-2005 Shared Task:** Semantic Role Labeling. Proceedings of CoNLL-2005. Ann Arbor, Michigan: ACL. 2005. p. 152-164.

CASELI, H. M. et al. Building a Brazilian Portuguese parallel corpus of original and simplified texts. **Advances in Computational Linguistics, Research in Computer Science**, v. 41, p. 59-70, 2009.

COLLINS, M.; KOO, T. Discriminative Reranking for Natural Language Parsing. **Computational Linguistics**, v. 31, n. 1, p. 25-69, 2005.

CRAMMER, K.; SINGER, Y. On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines. **Journal of Machine Learning Research**, v. 2, p. 265–292, 2001.

DIETTERICH, T. G.; BAKIRI, G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. **Journal of Artificial Intelligence Research**, v. 2, p. 263–286, 1995.

DUAN, K. B.; KEERTHI, S. S. Which Is the Best Multiclass SVM Method? An Empirical Study. Multiple Classifier Systems. **Lecture Notes in Computer Science**, v. 3541, p. 278–285, 2005.

DURAN, M. S.; ALUÍSIO, S. M. **Propbank-Br:** a Brazilian Treebank annotated with semantic role labels. Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012). Istanbul: European Language Resources Association. 2012.

FAN, R.-E. et al. Liblinear: A Library for Large Linear Classification. **Journal of Machine Learning Research**, v. 9, p. 1871-1874, 2008.

FILLMORE, C. J. **"The Case for Case". In Bach and Harms**. New York: Holt, Rinehart & Winston, 1968.

FILLMORE, C. J. Frames and the Semantics of Understanding. **Quaderni di Semantica**, v. 6, n. 2, p. 222-254, 1985.

FLEISCHMAN, M.; KWON, N.; HOVY, E. **Maximum entropy models for FrameNet classification**. Proceedings of the 2003 conference on Empirical methods in natural language processing. Morristown, USA: ACL. 2003. p. 49-56.

FONSECA, E. R. **Uma abordagem conexionista para anotação de papéis semânticos**. Universidade de São Paulo – Instituto de Ciências Matemáticas e de Computação. São Carlos. 2013.

GILDEA, D.; JURAFSKY, D. Automatic labeling of semantic roles. **Computational Linguistics**, v. 28, n. 3, p. 245-288, 2002.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Online: MIT Press, 2016.

GUYON, I.; ELISSEEFF, A. An Introduction to Variable and Feature Selection. **Journal of Machine Learning Research**, v. 3, 2003.

HAJICOVA, E.; PANEVOVA, J.; SGALL, P. **A Manual for Tectogrammatic Tagging of the Prague Dependency Treebank**. Charles University. Prague, Czech Republic. 2000. (UFAL/CKL Technical Report TR-2000-09).

HSU, C.-W.; LIN, C.-J. A Comparison of Methods for Multiclass Support Vector Machines. **IEEE Transactions on Neural Networks**, v. 13, n. 2, 2002.

JAMES, G. et al. **An Introduction to Statistical Learning**. New York: Springer, 2012.

JURAFSKY, D. Obituary Charles J. Fillmore. **Computational Linguistics**, v. 40, n. 1, p. 725-731, September 2004.

LIMA, M. C. P. B. A gramática dos casos e o "dativo". **Alfa**, São Paulo, v. 26, n. 1, p. 33-46, 1982.

MARCUS, M. P.; SANTORINI, B.; MARCINKIEWICZ, M. A. Building a Large Annotated. **Computational Linguistics**, v. 19, n. 2, p. 313

MÁRQUEZ, L. et al. **Semeval-2007 task 09:** Multilevel semantic annotation of catalan and Spanish. SemEval '07 Proceedings of the 4th International Workshop on Semantic Evaluations. Stroudsburg, PA, USA: ACL. 2007. p. 42-47.

MÀRQUEZ, L. et al. Semantic Role Labeling: An Introduction to the Special Issue. **Association for Computational Linguistics**, v. 34, n. 2, p. 145-159, 2008.

MCCOY, A. M. B. C. **A Case grammar classiication of Spanish verbs**. Xerox Univ. Microfilms. Ann Arbor. 1969.

MESQUITA, R. M.; MARTOS, C. R. Português - Linguagem & Realidade. 3. ed. São Paulo: Saraiva, v. 1, 1994. p. 26.

MORANTE, R.; BOSCH, A. V. D. Feature Construction for Memory-Based Semantic Role Labeling of Catalan and Spanish. **Recent Advances in Natural Language Processing V**, Amsterdam, v. 309, p. 131-142, 2009.

MOREDA, P. **Los Roles Semánticos en la Tecnología del Lenguaje Humano: Anotación y Aplicación**. Universidad de Alicante. Alicante, Spain. 2008.

MOTTA, E. N. **Indução e seleção incrementais de atributos no aprendizado supervisionado**. Pontifícia Universidade Católica do Rio de Janeiro - Departamento de Informática. Rio de Janeiro. 2014.

NILSEN, D. L. G. Toward a semantic specfication of deep case. **Janus linguarum, Series Minor**, Mouton Publishers, The Hague, v. 152, 1972.

PALMER, M.; KINGSBURY, P.; GILDEA, D. The Proposition Bank: An Annotated Corpus of Semantic Roles. **Computational Linguistics**, v. 31, n. 1, p. 71–106, 2005.

PLATT, J. **Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines**. Microsoft Research. Technical Report. 1998.

PLATT, J.; CRISTIANINI, N.; SHAWE-TAYLOR, J. Large margin DAGs for multiclass classification. **Advances in Neural Information Processing Systems**, MIT Press, p. 547–553, 2000.

PRADHAN, S. S.; WARD, W.; MARTIN, J. H. Towards Robust Semantic Role Labeling. **Computational Linguistics**, v. 34, n. 2, p. 289-310, 2008.

ROCHA, P. A.; SANTOS, D. **CETEMPúblico:** Um corpus de grandes dimensões de linguagem jornalística portuguesa. V Encontro para o processamento computacional da língua portuguesa escrita e falada (PROPOR 2000). Atibaia: ICMC/USP. 2000. p. 131-140.

ROTH, M.; WOODSEND, K. **Composition of Word Representations Improves Semantic Role Labelling**. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics. 2014. p. 407–413.

SANTO, D.; BICK, E.; AFONSO, S. **Floresta Sintá(c)tica:** apresentação e história do projecto. Encontro Um passeio pela Floresta Sintá(c)tica. Coimbra, Portugal: Linguateca. 2007.

SEQUEIRA, J.; GONÇALVES, T.; QUARESMA, P. Semantic Role Labeling for Portuguese–A Preliminary Approach–. **Lecture Notes in Artificial Intelligence**, Heidelberg, v. 7243, p. 193-203, 2012.

SURDEANU et al. **The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies**. Proceedings of the Twelfth Conference on Computational Natural Language Learning. Manchester: Association for Computational Linguistics. 2008. p. 159-177.

TAULÉ, M. et al. **Mapping syntactic functions into semantic roles**. Proceedings of TLT2005. Barcelona, Spain: [s.n.]. 2005. p. 185-194.

TOUTANOVA, K.; HAGHIGHI, A.; MANNING, C. D. **Joint Learning Improves Semantic Role Labeling**. Proceedings of ACL-05. Ann Arbor, Michigan: ACM. 2005. p. 589-596.

TOUTANOVA, K.; HAGHIGHI, A.; MANNING, C. D. A Global Joint Model for Semantic Role Labeling. **Computational Linguistics**, v. 34, n. 2, p. 161-191, 2008.

VAPNIK, V. N. **The Nature of Statistical Learning Theory**. New York: Springer-Verlag, 1995.

XUE, N.; PALMER, M. **Calibrating Features for Semantic Role Labeling**. Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2004). Barcelona: John Benjamins. 2004. p. 88-94.