DISSERTAÇÃO DE MESTRADO

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM

CIÊNCIA DA COMPUTAÇÃO

"Aprendizado Sem-fim de Paráfrases"

ALUNO: PAULO CÉSAR POLASTRI
ORIENTADORA: Profa. Dra. HELENA DE MEDEIROS CASELI
CO - ORIENTADORA: Profa. Dra. ELOIZE ROSSI MARQUES SENO

São Carlos
Fevereiro/2016

CAIXA POSTAL 676
FONE/FAX: (16) 3351-8233
13565-905 - SÃO CARLOS - SP
BRASIL

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

APRENDIZADO SEM-FIM DE PARÁFRASES

PAULO CÉSAR POLASTRI

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Inteligência Artificial.
Orientadora: Dra. Helena de Medeiros Caseli

São Carlos - SP
Fevereiro/2016

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

Polastri, Paulo César
P762a Aprendizado sem-fim de paráfrases / Paulo César Polastri. -- São Carlos : UFSCar, 2016.
113 p.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2016.

1. Paráfrases. 2. Reconhecimento automático de paráfrases. 3. Aprendizado de máquina sem-fim. 4. Processamento de língua natural. 5. Português do Brasil. I. Título.

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a defesa de dissertação de mestrado do candidato Paulo Cesar Polastri, realizada em 04/03/2016.

Profa. Dra. Helena de Medeiros Caseli
(UFSCar)

Profa. Dra. Heloisa de Arruda Camargo
(UFSCar)

Prof. Dr. Arnaldo Cândido Júnior
(UTFPR)

Certifico que a sessão de defesa foi realizada com a participação à distância do membro Prof. Dr. Arnaldo Cândido Júnior e, depois das arguições e deliberações realizadas, o participante à distância está de acordo com o conteúdo do parecer da comissão examinadora redigido no relatório de defesa do aluno Paulo César Polastri.

Profa. Dra. Helena de Medeiros Caseli
Presidente da Comissão Examinadora
(UFSCar)

Dedico aos meus pais, meu irmão e minha namorada.

AGRADECIMENTO

A Deus, por essa oportunidade, por me dar força e por me guiar pelos caminhos que me levaram a concluir essa importante etapa da minha vida.

Aos meus pais, por estarem do meu lado e sempre me apoiarem, seja nos momentos fáceis ou nos momentos difíceis.

A minha namorada, pelo apoio incondicional e pela compreensão.

Ao meu irmão, pelos conselhos ao longo deste período.

As minhas orientadoras, pela paciência, dedicação e valiosas orientações durante o longo desses anos.

A todos os professores pela ajuda valiosa.

"Mudam-se os tempos, mudam-se as vontades,
Muda-se o ser, muda-se a confiança;
Todo o mundo é composto de mudança,
Tomando sempre novas qualidades".

Luís de Camões

RESUMO

Usar palavras diferentes para expressar/transmitir a mesma mensagem é uma necessidade em qualquer língua natural e, como tal, deve ser investigada nas pesquisas em Processamento de Língua Natural (PLN). Quando se trata apenas de uma palavra simples, dizemos que as palavras intercambiáveis são *sinônimos*; enquanto o termo *paráfrase* é utilizado para expressar uma ideia mais geral e que pode envolver também mais de uma palavra. Por exemplo, as sentenças "o sinal está vermelho" e "o semáforo está fechado" são exemplo de paráfrases enquanto "sinal" e "semáforo" representam sinônimos, nesse contexto. O tratamento adequado de paráfrases é importante em diversas aplicações de PLN, como na Tradução Automática, onde paráfrases podem ser utilizadas para aumentar a cobertura de sistemas de Tradução Automática Estatística; na Sumarização Multidocumento, onde a identificação de paráfrases permite o reconhecimento de informações repetidas; e na Geração de Língua Natural, onde a geração de paráfrases permite criar textos mais variados e fluentes. O projeto descrito neste documento visa verificar se é possível aprender, de modo incremental e automático, paráfrases em nível de palavras a partir de corpus paralelo bilíngue, utilizando a estratégia de Aprendizado de Máquina Sem-fim (AMSF) e a Internet como fonte de conhecimento. O AMSF é uma estratégia de Aprendizado de Máquina, baseada na forma como os humanos aprendem: o que é aprendido previamente pode ser utilizado para aprender informações novas e talvez mais complexas, futuramente. Para tanto, o AMSF foi aplicado juntamente com a estratégia para a extração de paráfrases proposta por Bannard e Callison-Burch (2005) onde, a partir de corpus paralelo bilíngue, paráfrases são extraídas utilizando um idioma pivô. Nesse contexto, foi desenvolvido o NEPaL (*Never-Ending Paraphrase Learner*), sistema de AMSF responsável por: (1) extrair textos da internet, (2) alinhar os textos utilizando um idioma pivô, (3) classificar as candidatas de acordo com um modelo de classificação e (4) utilizar o conhecimento para produzir um novo modelo classificador e, consequentemente, adquirir mais conhecimento reiniciando o ciclo de aprendizado sem-fim.

Palavras-chave: paráfrases, reconhecimento automático de paráfrases, aprendizado de máquina sem-fim, processamento de língua natural, português do Brasil.

ABSTRACT

Use different words to express/convey the same message is a necessity in any natural language and, as such, should be investigated in research in Natural Language Processing (NLP). When it is just a simple word, we say that the interchangeable words are synonyms; while the term paraphrase is used to express a more general idea and that also may involve more than one word. For example, the sentences "the light is red" and "the light is closed" are examples of paraphrases as "sign" and "traffic light" represent synonymous in this context. Proper treatment of paraphrasing is important in several NLP applications, such as Machine Translation, which paraphrases can be used to increase the coverage of Statistical Machine Translation systems; on Multidocument Summarization, where paraphrases identification allows the recognition of repeated information; and Natural Language Generation, where the generation of paraphrases allows creating more varied and fluent texts. The project described in this document is intended to verify that is possible to learn, in an incremental and automatic way, paraphrases in words level from a bilingual parallel corpus, using Never-Ending Machine Learning (NEML) strategy and the Internet as a source of knowledge. The NEML is a machine learning strategy, based on how humans learn: what is learned previously can be used to learn new information and perhaps more complex in the future. Thus, the NEML has been applied together with the strategy for paraphrases extraction proposed by Bannard and Callison-Burch (2005) where, from bilingual parallel corpus, paraphrases are extracted using a pivot language. In this context, it was developed NEPaL (Never-Ending Paraphrase Learner) AMSF system responsible for: (1) extract the internet texts, (2) align the text using a pivot language, (3) rank the candidates according to a classification model and (4) use the knowledge to produce a new classifier model and therefore gain more knowledge restarting the never-ending learning cycle.

Keywords: paraphrase lexicon, automatic paraphrase recognition, never-ending machine learning, natural language processing, Brazilian Portuguese.

.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

PLN – *Processamento de Língua Natural*

NLP – *Natural Language Processing*

TA – *Tradução Automática*

NELL – *Never-Ending Language Learning*

NEPaL – *Never-Ending Praphrase Learner*

NEBEL – *Never-Ending Bilingual Equivalent Learner*

PorTAl – *Portal de Tradução Automática*

MT – *Machine Translation*

AM – *Aprendizado de Máquina*

AMSF – *Aprendizado de Máquina Sem-fim*

LALIC – Laboratório de Linguística e Inteligência Computacional

PoS – *Part-of-Speech*

SMT – *Statistical Machine Translation (Tradução Automtrica Estatística)*

PBSMT – *Phrase-Based Statistical Machine Translation*

PPDB – *ParaPhrase DataBase*

MTC – *Multiple-Translation Chinese Corpus*

TeP – *Thesaurus Eletrônico do Português do Brasil*

DSBWA – *Similaridade distribucional baseada no alinhamento lexical*

DSBSR – *Similaridade distribucional baseada em relações sintáticas*

EBMT – *Example-Based Machine Translation*

ARFF – *Attribute-Relation File Format*

LCSR – *Longest Common Subsequence Ratio*

NEPaLE – *Interface de anotação e avaliação do NEPaL*

NILC – *Núcleo Interinstitucional de Linguística Computacional*

SUMÁRIO

Capítulo 1

INTRODUÇÃO

Este capítulo tem o propósito de introduzir ao leitor conceitos sobre paráfrases e também apresentar as motivações que influenciaram este projeto e o objetivo do mesmo.
Este capítulo está organizado da seguinte forma. Na Seção 1.1, será apresentada uma breve introdução sobre paráfrases. Na Seção 1.2, serão apresentados os motivos que levaram a esta proposta. Na Seção 1.3, serão apresentados os objetivos da pesquisa. Por fim, a seção 1.4 apresenta como este trabalho está organizado.

1.1 Introdução

Reconhecer e extrair paráfrase são tarefas importantes para várias áreas do PLN, mas são pouco comuns quando o idioma alvo é o Português do Brasil. As paráfrases podem ser utilizadas, por exemplo, para gerar textos mais fluentes e auxiliar no reconhecimento de informações.

A relação entre duas palavras, quando apresentam o mesmo significado dentro do contexto, é chamada de paráfrase lexical (BARZILAY e MCKEOWN, 2001; BANNARD e CALLISON-BURCH, 2005). Nesse trabalho, assume-se que paráfrases são formas alternativas de se transmitir uma informação, utilizando diferentes expressões linguísticas. Na Tabela 1.1 são apresentados alguns exemplos de paráfrases formuladas de diversas maneiras, como, por exemplo: através do emprego de sinônimos, como em (1); mudança de voz ativa para voz passiva, como em (2); e também com a transformação de um discurso direto para indireto (ou vice-versa), como em (3).

As paráfrases são normalmente classificadas como paráfrases lexicais e paráfrases sintáticas. As paráfrases lexicais, assim denominadas por alguns autores como Barzilay e McKeown (2001), são aquelas formadas pela substituição de palavras por palavras/expressões equivalentes, como no exemplo (1) e (5) da Tabela 1.1, onde, em alguns casos, ”barrar” e “bloquear”, “choro” e ”tristeza” transmitem a mesma informação. Por sua vez, as paráfrases sintáticas ocorrem quando existe uma mudança na estrutura sintática, como, por exemplo, mudança de voz ativa e passiva (como no exemplo (2) da Tabela 1.1). Em alguns casos, como no exemplo (4) da Tabela 1.1, as paráfrases são lexicais e sintáticas ao mesmo tempo.

Tabela 1.1 - Exemplos de paráfrases.

(1)	barrar bloquear
(2)	A Honda construiu outro carro Outro carro foi construído pela Honda
(3)	Então, Marcelo disse: -Eu serei paciente Então, Marcelo disse que seria paciente
(4)	Letícia vendeu uma bota para Bia. Bia comprou uma bota de Letícia.
(5)	choro tristeza
(6)	Agulhas negras Academia militar agulhas negras

Segundo Barzilay e McKeown (2001), as paráfrases podem ainda ocorrer em três diferentes níveis de granularidade: (i) palavras, ocorrendo entre palavras simples, (ii) sintagmas, ocorrendo entre grupos de palavras (ou entre uma palavra simples e um grupo de palavras), e (iii) sentenças. Exemplos desses três níveis de paráfrases são apresentados na Tabela 1.1, onde (1) é um exemplo de paráfrases em nível de palavras, (6), em nível de sintagmas e (3), em nível de sentença.

Neste projeto o foco está no reconhecimento automático de paráfrases lexicais. Esse tema já foi bastante estudado na literatura. Entre os métodos pesquisados estão os que utilizam corpus paralelo, mesma estratégia adotada neste projeto. O uso do corpus paralelo permite equacionar alinhamentos em um determinado idioma alvo a partir de alinhamentos paralelos com outro idioma (usado como pivô), como apresentado em Bannard e Callison-Burch (2005).

Atualmente, os trabalhos que lidam com o tratamento de paráfrases disponíveis na literatura se dividem em três tarefas: i) o reconhecimento de paráfrases (SENO e NUNES, 2009; BARZILAY e MCKEOWN, 2005); ii) a extração

de paráfrases (BARZILAY e MCKEOWN, 2001; AZIZ e SPECIA, 2013); e iii) a geração de paráfrases (PANG et al., 2003; QUIRK et al., 2004; ZHAO et al., 2010).

O reconhecimento (ou identificação) de paráfrases é uma tarefa que consiste em determinar se duas ou mais expressões são ou não uma paráfrase. Essa tarefa pode ser útil em algumas aplicações, como a Sumarização Automática, auxiliando no reconhecimento de informações redundantes.

A extração de paráfrase é uma tarefa que necessita que as paráfrases sejam reconhecidas previamente. Após o reconhecimento, as paráfrases são extraídas para uso posterior, como a criação de bases de dados de frases, podendo também serem aplicadas em sistemas de tradução automática estatística, e serem úteis para aumentar a cobertura[1] do texto, no caso de idiomas com corpora muito pequenos (BANNARD e CALLISON-BURCH 2005).

Já a geração de paráfrases é uma tarefa que consiste em gerar, a partir de uma sentença de entrada (ou de um *template*), o maior número de expressões ou *templates* possível. Essa tarefa é importante para a Geração de Língua Natural, auxiliando na geração de textos variados.

Em alguns dos trabalhos disponíveis na literatura, a identificação de paráfrases é feita através de medidas de similaridade como: similaridade de contexto (BARZILAY e MCKEOWN, 2001), similaridade lexical (VAN DER PLAS E TIEDEMANN, 2006), similaridade lexical e sintática (PANG et al., 2003) e similaridade lexical, sintática e semântica (SENO e NUNES, 2009).

O projeto de mestrado aqui descrito visa à identificação de paráfrases lexicais com a posterior extração das mesmas para realimentar o sistema de aprendizado sem-fim e gerar um léxico de paráfrases que será disponibilizado para toda a comunidade científica. Essa identificação foi realizada a partir de textos escritos em português do Brasil e traduzidos para o inglês, por jornalistas do jornal Folha de São Paulo[2], em notícias da versão *online* do jornal Folha de São Paulo. Para tanto, utilizou-se o método proposto por Bannard e Callison-Burch (2005) juntamente com a estratégia de Aprendizado de Máquina Sem-Fim (AMSF).

Bannard e Callison-Burch (2005) propõem um método de extração de paráfrases que utiliza um idioma como pivô. Em tal método, as paráfrases são

[1] A Cobertura representa o total de paráfrases aprendidas dividido pelo total de equivalentes que poderiam ser aprendidas no corpus.

[2] Disponível em: http://www.folha.uol.com.br/. Acesso em: 21/01/2016.

identificadas por meio de alinhamento lexical de um corpus paralelo bilíngue. Experimentos realizados pelos autores comprovaram que em 48,9% dos casos avaliados as paráfrases extraídas com base em alinhamento lexical automático apresentavam equivalência semântica.

O AMSF é uma estratégia de aprendizado de máquina que visa à obtenção de conhecimento de forma incremental e constante, acumulando cada vez mais conhecimento com o tempo. Com essa estratégia, o conhecimento acumulado é usado novamente para aprender algo novo e, inclusive, pode ajudar a melhorar a capacidade de aprendizado. A escolha dessa estratégia foi devido à recentes aplicações que a utilizam como um meio de extrair conhecimento do grande volume de informação disponível, principalmente na *web*. Essa estratégia vem se mostrando eficaz no projeto *ReadtheWeb*[3], que acumula mais de 50 milhões de crenças aprendidas. No LALIC (Laboratório de Linguística e Inteligência Computacional), laboratório no qual este projeto foi desenvolvido, outros aprendizes sem-fim já foram produzidos com resultados disponibilizados em: http://www.lalic.dc.ufscar.br/never-ending/ (MITCHELL, 2008).

Os resultados obtidos neste projeto de mestrado foram utilizados para criar um léxico de paráfrases (ou uma base de dados), uma atividade que, quando desenvolvida manualmente, é custosa e depende de mão de obra de especialistas. Uma das motivações para o desenvolvimento desse recurso é melhorar o desempenho de tradutores automáticos, além de outros projetos, como o PorTAl[4] (VIEIRA e CASELI, 2011), podendo oferecer melhores opções de tradução e maior variedade textual.

Além dos textos extraídos da versão *online* do jornal Folha de São Paulo, outros recursos foram utilizados neste projeto como: alinhadores de sentenças do PorTAl, alinhador lexical GIZA++ (OCH e NEY, 2003) através do *toolkit* de tradução automática Moses[5] (KOEHN et al., 2007); o etiquetador morfossintático, também disponível no PorTAl , além do *toolkit* de aprendizado de máquina Weka[6] (HALL et al., 2009).

[3] Disponível em: http://rtw.ml.cmu.edu/rtw/. Acesso em: 22/05/2014.
[4] Disponível em:http://www.lalic.dc.ufscar.br/portal/. Acesso em: 21/01/2016.
[5] Disponível em: http://www.statmt.org/moses. Acesso em: 12/01/2016.
[6] Disponível em: http://www.cs.waikato.ac.nz/ml/weka/. Acesso em: 12/01/2016.

1.2 Motivação

As paráfrases são importantes em diversas aplicações em PLN, como em sistemas de Perguntas e Respostas, onde a ocorrência de paráfrase entre a resposta fornecida pelo usuário e a resposta correta pode ser um indício de que a resposta do usuário também está correta (IBRAHIM et al., 2003). Na sumarização multidocumento[7], o reconhecimento de paráfrases é útil para identificar informações redundantes nos documentos de entrada (BARZILAY e MCKEOWN, 2001; BARZILAY et al., 2002). Na geração de língua natural, as paráfrases são utilizadas para criar textos mais variados e fluentes (IORDANSKAJA et al., 1991). Na tradução automática, ajuda a criar traduções mais claras e precisas, aumentando a cobertura do texto, ao substituir uma palavra/sintagma fonte desconhecida por uma paráfrase, e realizando a tradução para a língua alvo a partir dessa paráfrase (CALLISON-BURCH et al., 2006). Na fusão automática de sentenças[8], paráfrases auxiliam na criação de textos mais completos, com a união de sentenças[9], e também mais objetivos, com a intersecção de sentenças[10], sendo importantes na identificação de informações pleonásticas/repetidas[11] (SENO e NUNES, 2009).

1.3 Objetivos

O projeto descrito neste documento teve como objetivo:

"*Verificar se é possível utilizar a estratégia de aprendizado sem-fim e a Internet para aprender, de modo incremental e automático, conhecimento útil para a identificação e a extração de paráfrases."*

[7] Sumarização multidocumento consiste em produzir um único sumário a partir de textos referentes ao mesmo tema ou temas relacionados (MANI, 2001).
[8] Fusão de sentenças consiste na produção de uma única sentença que contenha informações comuns a partir de um grupo de sentenças relacionadas (SENO e NUNES, 2009).
[9] A união de sentenças consiste em preservar todas as informações de um grupo de sentenças, nas sentenças resultantes (SENO e NUNES, 2009).
[10] A intersecção de sentenças consiste em manter apenas informações comuns ao grupo de sentenças resultantes (SENO e NUNES, 2009).
[11] Informação pleonástica é uma informação que se repete em um determinado contexto.

Para alcançar esse objetivo, foi empregada a estratégia de aprendizado de máquina sem-fim, assim como no NELL (*Never-Ending Language Learning)*, sistema de aprendizado de máquina sem-fim do projeto *ReadtheWeb*[12].

O sistema de aprendizado sem-fim construído neste projeto, nomeado NEPaL (*Never-Ending Paraphrase Learner),* foi desenvolvido com o intuito de aplicar, aos textos coletados da Internet, o método de extração de paráfrases proposto por Bannard e Callison-Burch (2005), gerando, assim, de modo incremental, um léxico de paráfrases. Esse método foi escolhido porque apresenta uma estratégia alternativa de extração de paráfrases. Até então, a maioria dos métodos apresentados visa extrair paráfrases utilizando corpus monolíngue. Segundo Bannard e Callison-Burch (2005), é possível obter bons resultados quanto à extração de paráfrases utilizando um corpus paralelo bilíngue, além desse ser um recurso mais comum de ser encontrado.

1.4 Organização do Texto

O texto dessa dissertação está organizado como descrito a seguir.

O Capítulo 2 apresenta de forma mais detalhada definições de paráfrases e sinônimos, as classificações empregadas e quais suas principais aplicações no Processamento de Língua Natural, de acordo com os principais trabalhos na área disponíveis na literatura. Ainda no Capítulo 2 são apresentados os principais trabalhos que descrevem tarefas de identificação*,* extração e geração de paráfrases e também reconhecimento e extração de sinônimos.

O Capítulo 3 detalha o desenvolvimento e funcionamento do sistema criado neste projeto, o *Never-Ending Paraphrase Learner* (NEPaL), assim como os recursos e ferramentas que foram utilizados no desenvolvimento desse sistema.

No Capítulo 4, são descritos os experimentos realizados e também os resultados obtidos, tanto em relação às paráfrases aprendidas quanto em relação a evolução do aprendizado sem-fim. Por fim, o Capítulo 5 traz as conclusões e considerações finais deste documento.

[12] Disponível em: http://rtw.ml.cmu.edu/rtw/. Acesso em: 22/05/2014.

Capítulo 2

REVISÃO BIBLIOGRÁFICA

Este capítulo tem o propósito de contextualizar o leitor sobre os principais conceitos, estratégias e métodos de extração de paráfrases e também de sinônimos, além de apresentar alguns trabalhos disponíveis na literatura.

Esse capítulo está organizado da seguinte forma. Na Seção 2.1 são apresentados os conceitos de paráfrases. Na Seção 2.2, são apresentados alguns trabalhos sobre o reconhecimento, a extração e a geração de paráfrases e a extração de sinônimos, relacionados a esta pesquisa. Por fim, a Seção 2.3 resume o conteúdo discutido neste capítulo.

2.1 Paráfrases

Paráfrases são formas alternativas de se transmitir uma informação, utilizando diferentes expressões linguísticas (BANNARD e CALLISON BURCH, 2005; BARZILAY e MCKEOWN, 2001; AZIZ e SPECIA, 2013). Segundo Halliday (1985 apud BARZILAY e MCKEOWN, 2001), as paráfrases apresentam equivalência conceitual aproximada, como, por exemplo, na relação entre as palavras “tricolores” e “gremistas”, onde não existe relação de sinonímia, porém, em certos casos, pode-se afirmar que são paráfrases, dependendo do contexto onde estão inseridas.

De acordo com BARZILAY e MCKEOWN (2001), as paráfrases são classificadas como lexical e sintática, como nos exemplos (3) e (4) apresentados na Tabela 2.1, respectivamente. Segundo as autoras, elas podem ocorrer em três diferentes níveis de granularidade: palavras, quando envolvem duas palavras; sintagmas, quando envolvem grupos de palavras (podendo ocorrer entre um grupo de palavras e uma única palavra); e sentenças, quando envolvem uma sentença

completa (de uma pontuação delimitadora a outra, como ponto final, exclamação ou interrogação). Também são apresentados alguns exemplos de paráfrases em nível de palavras, (exemplo (5)), sintagmas, (exemplos (6), (7) e (8)) e sentenças (exemplos (9), (10) e (11)). As paráfrases podem, ainda, ser atômicas, como em (8), quando formadas por palavras que isoladamente não constituem uma paráfrase da outra, como, por exemplo, “escândalo” e “compra de votos” constituem paráfrases dependendo do contexto onde estão inseridas, e compostas, como em (9), quando podem ser decompostas em outras paráfrases (BARZILAY e MCKEOWN, 2001).

Tabela 2.1 - Exemplos de paráfrases.

(1)	Ele me ajudou a resolver o problema. Ele me prestou auxilio para solucionar o empasse.
(2)	Os Tucanos pretendem se eleger novamente. Os Políticos do PSDB pretendem se eleger novamente.
(3)	punição castigo
(4)	Meu pai fez o almoço. O almoço foi feito pelo meu pai.
(5)	resolver solucionar
(6)	Ajudar Prestar auxílio
(7)	comida chinesa culinária típica da China
(8)	escândalo do Mensalão compra de votos de parlamentares
(9)	Hoje de manhã, atletas do Catanduvense treinaram em Itajobi. Os jogadores de Catanduva treinaram em Itajobi agora de manhã.
(10)	Os casos de dengue aumentaram este ano. Aumentaram, este ano, os casos de dengue.
(11)	O jogador anunciou que se aposenta no final deste ano. O jogador disse que no fim do ano, se aposentaria.

Conforme dito no Capítulo 1, o foco do trabalho foi a extração de paráfrases lexicais (em nível de palavras, portanto).

Os autores dos trabalhos da literatura atual sobre paráfrases focam suas pesquisas em três tarefas principais: no reconhecimento (ou identificação) (MALAKASIOTIS, 2009; BARZILAY e MCKEOWN, 2005), na extração (BARZILAY e MCKEOWN, 2001; BANNARD e CALLISON-BURCH, 2005; AZIZ e SPECIA,

2013) e na geração de paráfrases (QUIRK et al., 2004; PANG et al., 2003; ZHAO et al., 2010).

Os métodos de reconhecimento de paráfrases julgam se duas expressões formam um par de paráfrases ou não.

Diferentemente dos métodos de reconhecimento de paráfrase, os métodos de geração de paráfrase, por sua vez, recebem uma única expressão (ou *template*, como, por exemplo, "X is author of Y") de linguagem como entrada, e a partir da entrada geram o maior número de expressões ou modelos possíveis. Essas sentenças geradas constituem paráfrases (ANDROUTSOPOULOS e MALAKASIOTIS, 2010).

Quando as expressões de entrada e saída são *templates*, a aplicação de um método *bootstrapping*[13] é viável, pois esses *templates* podem ser empregados para extrair novos exemplos, e o método pode ser empregado em grandes corpora, como a *web*, usando sementes[14] e extraindo paráfrases, pois, por exemplo, através do modelo "X é o autor de Y" e da semente "X = J.R.R. Tolkien, Y = The Lord of the rings", é possível encontrar as sentenças "J.R.R. Tolkien escreveu The Lord of the Rings" e "The Lord of the Rings foi escrito por J.R.R. Tolkien", e a partir dessas sentenças criar os modelos "X escreveu Y" e "Y foi escrito por X".

Os métodos de extração de paráfrase geralmente processam grandes corpora para identificar e extrair paráfrases, e depois de reconhecidas as correspondências são armazenadas, podendo ser usadas para construir recursos como bases de dados de frases ou conjunto de regras.

É importante salientar que, para realizar a extração de paráfrases, é necessário previamente que esta seja reconhecida, ou seja, qualquer método que visa à extração de paráfrases depende também do processo de reconhecimento.

A pesquisa aqui descrita teve como foco a extração de paráfrases a partir da *web*. Vários métodos de extração e reconhecimento de paráfrases foram estudados e os trabalhos pesquisados são apresentados em detalhes nas próximas seções.

Os métodos estudados utilizam diferentes tipos de corpus em seu processamento. Os corpora disponíveis que foram pesquisados e avaliados durante este projeto são classificados da seguinte forma:

[13] Bootstrapping é um algoritmo que utiliza os resultados obtidos anteriormente como modelos para encontrar novos resultados.
[14] Sementes são exemplos de treinamento rotulados que são utilizados para treinar um modelo.

- Corpus Monolíngue: Consiste em uma coleção de textos em apenas um idioma. Entre alguns corpora monolíngues estão o British National Corpus[15], em inglês, contendo centenas de milhões de palavras. Esse tipo de corpus foi usado, por exemplo, no trabalho de Barzilay e McKeown (2001) (inglês).
- Corpus Bilíngue ou Multilíngue: Este corpus é formado necessariamente por textos em dois ou mais idiomas. Um exemplo de corpus multilíngue é o corpus FAPESP (AZIZ e SPECIA, 2011), com cerca de 4 milhões palavras em português do Brasil, que é formado por textos originais da revista Pesquisa FAPESP, escritos em português do Brasil, e por traduções para o inglês e o espanhol. Um corpus bilíngue foi utilizado no trabalho de Bannard e Callison-Burch (2005) (inglês-alemão) e um corpus multilíngue foi utilizado no trabalho de Van Der Plas e Tiedemann (2006), composto por onze idiomas diferentes.
- Corpus Comparável: É formado por conjuntos de textos em uma ou mais línguas, que se concentram em um assunto em particular, mas não tratam de traduções literais. Esse tipo de corpus pode ser formado, por exemplo, por artigos jornalísticos que relatam um mesmo acontecimento, mas que são produzidos por diferentes agências de notícias da *web*. Um exemplo de corpus comparável em português do Brasil é o CSTNews (ALEIXO e PARDO, 2008). Um corpus desse tipo foi utilizado no trabalho de Seno e Nunes (2009), em português. Um corpus comparável pode ser monolíngue, bilíngue ou multilíngue.
- Corpus Paralelo: É formado por grupos de textos em uma ou várias línguas. Nesse tipo de corpus, existe necessariamente uma relação de tradução entre os textos. Pode ser monolíngue ou multilíngue. Corpora paralelos podem ser formados, por exemplo, por traduções diferentes de uma obra para uma mesma língua (nesse caso, trata-se de um corpus paralelo monolíngue). Ou ainda, pode ser formado por textos em determinado idioma e sua correspondente tradução para outro idioma (nesse caso, denomina-se corpus paralelo bilíngue). Entre alguns exemplos de corpus paralelo bilíngue podemos citar o COMPARA[16] e o FAPESP (AZIZ e SPECIA, 2011), formados por textos paralelos em inglês e português. Um

[15] Disponível em: http://www.natcorp.ox.ac.uk/. Acesso em: 22/01/2016.
[16] Disponível em: http://www.linguateca.pt/COMPARA/. Acesso em: 22/01/2016.

corpus paralelo pode ser monolíngue, bilíngue ou multilíngue. Ganitkevitch et alli (2013) utiliza corpus paralelo (inglês-alemão).

Nesse trabalho, o corpus utilizado é paralelo bilíngue. Esse tipo de corpus foi escolhido porque é o mesmo utilizado por Bannard e Callison-Burch (2005) que inspirou esse trabalho.

2.2 Trabalhos Relacionados

Nesta seção são apresentados alguns trabalhos sobre reconhecimento, extração e geração de paráfrases que se mostraram mais relevantes para o desenvolvimento deste projeto. Todos os trabalhos, independentemente da tarefa que trata, são interessantes e foram estudados com o intuito de que ideias pudessem ser utilizadas neste projeto.

2.2.1 Barzilay e McKeown (2001)

O método de extração de paráfrases proposto por Barzilay e Mckeown (2001) é baseado em aprendizado não supervisionado e tem por objetivo extrair paráfrases lexicais e sintáticas em nível de palavras e sequências de palavras, a partir de textos paralelos monolíngue em inglês. Mais especificamente, as autoras usaram uma coleção de várias traduções para o inglês de um mesmo romance em francês. Um exemplo dessas traduções paralelas é mostrado na Figura 2.1

O método apresentado por Barzilay e McKeown (2001), a partir das traduções paralelas previamente alinhadas em nível de sentenças, como no exemplo da Figura 2.1, extrai pares de paráfrases, como "*burst into tears*" e "*cried*" e "*comfort" e "console".*

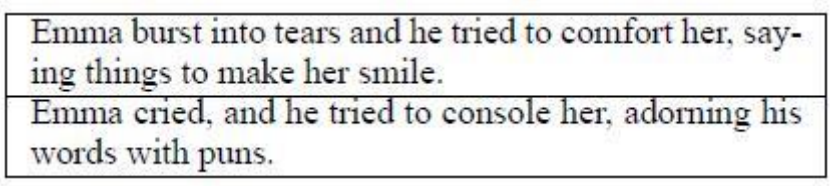


Figura 2.1 - Exemplos de paráfrases. Fonte: (BARZILAY e MCKEOWN, 2001, p. 1).

Para tanto, o método parte do pressuposto de que sintagmas em sentenças paralelas monolíngues alinhadas, encontradas em contextos similares, assim como no exemplo da Figura 2.1, são paráfrases. Além disso, o método também extrai padrões sintáticos, ou seja, *templates*, que são combinados usando medidas de correlação, como informações morfológicas e de *Part-of-Speech* (*PoS*[17]*)* comuns em Tradução Automática.

Em outras palavras, o método de Barzilay e McKeown (2001) se baseia nas semelhanças de contexto entre sentenças paralelas, partindo de um ponto comum entre elas, isto é, de um mesmo item lexical. A hipótese das autoras é de que em sentenças paralelas as paráfrases ocorrem entre contextos semelhantes. Padrões sintáticos (*templates)* como (VB NN)<->(VB) (ou seja, verbo e substantivo <-> verbo) são gerados a partir das paráfrases previamente extraídas e aplicados novamente ao corpus, sendo úteis para aprender novas paráfrases, continuamente, enquanto houver novas paráfrases a serem descobertas.

Dessa forma, a abordagem permite a identificação de paráfrases em nível de palavras e de sintagmas, além de extrair *templates*, produzindo um conjunto de padrões de paráfrases, baseando-se em informações morfológicas e uma marcação de *PoS*.

Neste trabalho foram utilizadas traduções de cinco livros, entre eles *Flaubert's Madame Bovary, Andersen's Fairy Tales* e *Verne's Twenty Thousand Leagues Under the Sea.* No total, o corpus contém 11 traduções. As traduções nunca são idênticas, variando de tradutor para tradutor, sendo, portanto, fonte natural de paráfrases (BARZILAY e MCKEOWN, 2001).

Antes do processamento do algoritmo as sentenças devem ser alinhadas em nível de sentenças. Esse alinhamento é realizado usando programação dinâmica (GALE e CHURCH, 1991) com uma função ponderada baseada na quantidade de palavras que são comuns no par de sentenças alinhadas.

A extração de paráfrases e *templates* é feita através das descrições léxicas e sintáticas do par de paráfrases. Cada palavra do contexto de palavras iguais (lexicalmente idênticas) em sentenças paralelas é marcada com etiquetas de *PoS*.

[17] PoS ou etiquetação de PoS trata da classificação de palavras quanto sua classe gramatical.

Nos experimentos descritos pelas autoras foram gerados 44.562 pares de sentenças alinhadas com 1.798.526 palavras. Para avaliar a precisão[18] do processo de alinhamento, foram analisados 127 pares de sentenças. Desses, 120 (94,5%) estavam corretos. Em seguida, as autoras usaram um etiquetador de PoS e um *chunker*[19] (MIKHEEV, 1997) para identificar classes como substantivos e verbos na sentença. Em seguida, palavras que aparecem nas duas sentenças de um par alinhado são utilizadas como sementes. Dessa forma, palavras idênticas que aparecem em um par de sentenças alinhadas são consideradas exemplos positivos de sementes, enquanto que palavras diferentes em sentenças alinhadas são consideradas exemplos negativos.

Conforme relatado pelas autoras, o algoritmo proposto produziu 9.483 pares de paráfrases lexicais e 25 *templates*. Destas paráfrases, 500 foram avaliadas manualmente com o intuito de verificar se os juízes humanos concordavam com os resultados do algoritmo. Os juízes receberam como orientação uma definição de que pares de paráfrase continham "equivalência conceitual aproximada" (BARZILAY e MCKEOWN, 2001). Para avaliar a influência do contexto na produção de paráfrases, os resultados foram avaliados de duas formas: (i) considerando o contexto e (ii) não considerando o contexto. Primeiramente, cada juiz humano avaliou um par de paráfrases isoladas (sem conhecer o contexto em que ocorreram), e depois da sua resposta, avaliou o mesmo par de paráfrases de acordo com o contexto original. Cada item foi avaliado por dois juízes e a concordância entre eles foi calculada por meio do coeficiente Kappa (κ) (CARLETTA, 1996).

O primeiro juiz considerou 439 pares de paráfrases (87,8%) como corretos, enquanto o segundo juiz considerou apenas 426 (85,2%), com valor *kappa* igual a κ=0,67 (CARLETTA, 1996). Na análise dos resultados para os julgamentos considerando o contexto obteve-se uma concordância ainda maior, κ=0,97, sendo avaliadas 459 (91,8%) e 457 (91,4%) paráfrases como corretas pelo primeiro e o segundo juízes, respectivamente.

Para tentar gerar uma estimativa da cobertura, foram realizadas algumas avaliações manuais, onde 50 sentenças foram utilizadas para extração de paráfrases por juízes humanos e, em seguida, contou-se quantas dessas paráfrases

[18] A Precisão representa o número de paráfrases aprendidas corretamente dividido pelo total de paráfrases aprendidas.
[19] *Chunker* é uma alternativa de análise que fornece uma estrutura sintática parcial de uma sentença.

foram identificadas pelo algoritmo. De um total de 70 paráfrases extraídas por humanos, 48 (69%) foram também encontradas pelo algoritmo.

O método de Barzilay e McKeown (2001) atinge ótimos percentuais tanto em precisão quanto em cobertura, porém utiliza corpus paralelo monolíngue, diferentemente do projeto aqui descrito, cujo método empregado utiliza corpus paralelo bilíngue, um recurso mais fácil de ser encontrado do que o monolíngue.

2.2.2 Bannard e Callison-Burch (2005)

Bannard e Callison-Burch (2005) extraem paráfrases em inglês a partir de corpora paralelos bilíngues, usando técnicas de alinhamento com base em modelos estatísticos (BROWN et al., 1993) bastante comuns na tradução automática estatística (SMT). Para a extração de paráfrases em inglês, os autores utilizam um idioma como pivô, no caso o alemão. Mais especificamente, o método proposto considera que um sintagma em um idioma A (alvo) pode ser traduzido de várias formas no idioma B (pivô) e cada uma dessas traduções no idioma B pode ser traduzido novamente para o idioma A, gerando sintagmas correspondentes (ou seja, paráfrases) no idioma A.

A essência desse método está em alinhar sintagmas em um corpus paralelo bilíngue e equacionar diferentes sintagmas no idioma alvo (inglês, nesse caso) que estão alinhadas com o mesmo sintagma no idioma pivô, conforme ilustrado na Figura 2.2.

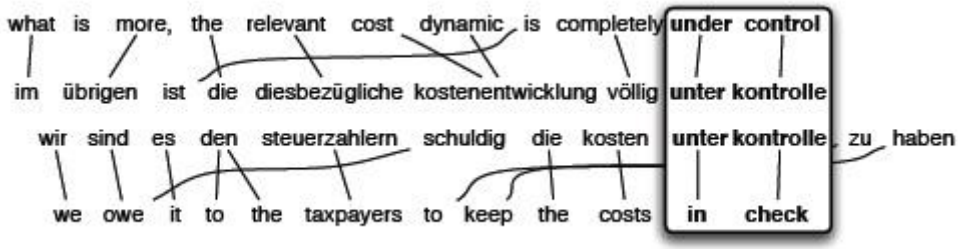


Figura 2.2 - Exemplo de alinhamento utilizando idioma pivô. Fonte: (BANNARD e CALLISON-BURCH, 2005, p. 3).

Como é possível notar pelo exemplo da Figura 2.2, o sintagma "*under control*" no idioma alvo (inglês) foi traduzida para o idioma pivô (alemão) como "*unter kontrolle*", para a qual também se tem o sintagma (em inglês) "*in check*". Dessa forma, através dos alinhamentos nos sentidos alvo-fonte e fonte-alvo, o método

classifica o par alinhado "*under control*" e "*in check*" como candidato a paráfrase, usando um modelo probabilístico.

Os alinhamentos entre sintagmas do idioma alvo (em inglês) são encontrados por intermédio dos alinhamentos com o idioma pivô (o alemão). A probabilidade de cada par de alinhamentos ser uma paráfrase é calculada de modo similar ao que ocorre na abordagem *Phrase-Based Statistical Machine Translation* (PBSMT). Na PBSMT, as palavras são agrupadas em sentenças e a probabilidade de uma sentença em um idioma alvo ser uma tradução de uma sentença em um idioma fonte é calculada com base nas probabilidades dos alinhamentos das palavras que formam essas sentenças. Para isso, é feito um somatório de todas as ocorrências de alinhamentos em nível de palavras nessas sentenças. Quanto maior o resultado deste somatório, maior a probabilidade da tradução (ou paráfrase, no caso desse método) estar correta, assim como na equação (1).

$$p(f|e) = \sum_a p(f, a|e) \quad (1)$$

Porém, desta forma, alguns candidatos a paráfrases, mesmo obtendo probabilidade alta, podem não ser adequados em alguns casos, já que essa medida de pontuação não considera o contexto onde os alinhamentos foram encontrados. Para tentar contornar esse problema, cada candidato pode ser reclassificado de acordo com as informações contextuais, intercambiando o par de paráfrases entre as sentenças onde ocorre.

A vantagem dessa abordagem em relação a abordagem de Barzilay e McKeown (2001), que extraem paráfrases de corpus paralelo monolíngue, é a abrangência de *tokens* que os corpora bilíngues trazem e, consequentemente, a maior variedade textual (BANNARD e CALLISON-BURCH, 2005).

Em experimentos apresentados por Bannard e Callison-Burch (2005), 46 candidatas a paráfrases que ocorreram várias vezes nas primeiras 50000 sentenças do corpus bilíngue escolhido (versão alemão-inglês do corpus Europarl, segunda versão) (KOEHN, 2002) foram extraídas de sentenças da WordNet, para serem comparadas com as paráfrases obtidas. Os alinhamentos lexicais foram gerados usando GIZA++[20] (OCH e NEY, 2003). Um alinhamento lexical de referência (*gold standard*) também foi gerado para o conjunto de sintagmas a serem parafraseados,

[20] GIZA++ é um alinhador lexical usado para o alinhamento de corpora.

para todas as ocorrências da sentença em inglês, por meio da correção manual do alinhamento automático.

Em seguida, foi realizada a avaliação da precisão de cada paráfrase extraída dos dados alinhados manualmente como também das paráfrases que obtiveram maior pontuação, extraídas com base no alinhamento automático. Como a escolha da melhor paráfrase é dependente do contexto, para cada grupo de candidatos à paráfrase, os juízes foram instruídos a substituir o sintagma original pela candidata a paráfrase em 2 a 10 sentenças, totalizando 289 conjuntos de avaliação com 1366 sentenças. Essas sentenças foram julgadas por dois juízes falantes nativos do inglês levando em consideração a preservação do sentido original da sentença e a preservação gramatical. De acordo com a medida *kappa* (CARLETTA, 1996), a concordância dos juízes foi de 0,605.

Os julgamentos foram feitos levando em consideração:

- Alinhamentos manuais;
- Alinhamentos automáticos;
- Múltiplos corpora, onde porções francês-inglês, espanhol-inglês e italiano-inglês do corpus Europarl (KOEHN, 2002) foram adicionadas ao corpus alemão-inglês em uma tentativa de aumentar o desempenho; e
- Controlando o sentido da palavra, através da aplicação de um modelo que limitava o alinhamento entre pares de candidatos apenas com o sintagma alvo correspondente, restringindo assim os candidatos à paráfrase em apenas candidatos que tivessem o mesmo sentido.

Os resultados obtidos são apresentados na Tabela 2.2:

Tabela 2.2 - Resultados da avaliação. Fonte: (BANNARD e CALLISON-BURCH, 2005, p. 6).

	Probabilidade de Paráfrase(%)	Significado Correto(%)
Alinhamentos Manuais	74.9	84.7
Alinhamentos Automáticos	48.9	64.5
Usando Vários Corpora	55.0	65.4
Controlando o Sentido da Palavra	57.0	70.4

A Tabela 2.2 mostra os resultados de um experimento feito através substituição de 289 sentenças geradas pelo processo por sentenças do *gold standard*. Nesse experimento, através do alinhamento manual, 74,9% das paráfrases preservavam o significado da sentença e estavam gramaticalmente

corretas. Ignorando a restrição de que as novas sentenças deveriam estar gramaticalmente corretas, essa porcentagem sobe para 84,7%. Contudo, o desempenho do método caiu consideravelmente quando o alinhamento manual foi substituído pelo alinhamento automático: sendo que apenas 48,9% das candidatas a paráfrases extraídas a partir do alinhamento automático estavam corretas e 64,5% dos casos preservaram o significado.

Segundo Bannard e Callison-Burch (2005), o método apresenta vantagens em relação a métodos que utilizam corpus paralelo monolíngue pelo fato de que ele cobre uma maior variedade de gêneros textuais, além de criar uma lista de paráfrases de alta qualidade com suas respectivas probabilidades. Porém, como visto pelos valores da Tabela 2.2, a qualidade das paráfrases extraídas é altamente dependente da qualidade dos alinhamentos de palavras.

O método empregado no trabalho de Bannard e Callison-Burch (2005) é o método que inspirou este trabalho.

2.2.3 Ganitkevitch et alli (2013)

No trabalho de Ganitkevitch et alli (2013) é descrita a criação de uma base de dados de paráfrases, denominada ParaPhrase DataBase, ou apenas PPDB, composta por paráfrases geradas através do alinhamento de corpus paralelo bilíngue inglês-alemão, também baseado no método empregado por Bannard e Callison-Burch (2005), conforme descrito na seção 2.2.2. Porém, no método empregado por Ganitkevitch et alli (2013), a probabilidade de paráfrase é calculada com base em *scores* obtidos pelo contexto onde as candidatas a paráfrase estão inseridas.

Em comparação com o método empregado por Bannard e Callison-Burch (2005), para o caso *"thrown into jail", o método também foi capaz de extrair vários itens lexicais, além de "imprisoned", como "arrested", "detained", "imprisoned", "incarcerated", "jailed", "locked up", "taken into custody" e "thrown into prison",* considerados corretos, além de paráfrases incorretas e ruídos do corpus. Em comparação com o método de Lin e Pantel (2001), baseado na extração de paráfrases em corpus monolíngue, para este mesmo caso, além de paráfrases consideradas como corretas, foram extraídas também casos que, apesar de apresentarem significados próximos, não são paráfrases de "*throw into jail"*, como:

"*began the trial of", "cracked down on", "interrogated", "prosecuted" e "ordered the execution of*".

Para a construção da PPDB, a pontuação de um par de palavras/sintagmas candidatos a paráfrase é calculada considerando-se diversos fatores, entre eles: (a) fatores contextuais baseados em n-gramas das sentenças mais frequentes do Google corpus (BRANTS e FRANZ, 2006; LIN et al., 2010) e do corpus Annotated Gigaword (NAPOLES et al., 2012); (b) fatores contextuais baseados nas *n* palavras à esquerda e à direita da palavra/sintagma; (c) posição da palavra/sintagma na sentença; e (d) lema, etiqueta de *PoS* da palavra/sintagma na sentença, de acordo com a estrutura frasal.

Segundo os autores, a versão inglesa da base de dados, chamada de PPDB:Eng possui mais de 220 milhões de paráfrases, sendo que 73 milhões são paráfrases em nível de sintagmas, 8 milhões são paráfrases lexicais e 140 milhões são *templates* sintáticos, criados através de regras formadas com base na estrutura sintática de sintagmas.

Já a versão em espanhol, PPDB:Spa é formada por 196 milhões de paráfrases incluindo paráfrases lexicais, em nível de sintagmas e sintáticas, além de outras versões em outros idiomas. Para a construção da versão em inglês da PPDB, PPDB:Eng, foram utilizados vários corpora paralelos bilíngues, em vários idiomas, como francês, checo, alemão, espanhol, chinês e árabe. No total, foram utilizados 22 idiomas pivôs diferentes para a criação da base de dados. Tudo isso resulta em um corpus com mais de 2 bilhões de palavras em inglês. Para a construção da versão espanhola, PPDB:Spa, utilizou-se um corpus de 355 milhões de palavras.

Quanto à qualidade dos resultados, 3 juízes julgaram 1900 pares de paráfrases em inglês escolhidos aleatoriamente e atribuíram uma nota entre 1 e 5 para cada par, sendo 5 o melhor. As avaliações foram feitas desconsiderando o contexto das paráfrases. O resultado das avaliações é mostrado em um gráfico expresso na Figura 2.3. Observa-se que a maioria dos dados foi avaliada entre 3 e 5 (abaixo no gráfico), considerando os diferentes limiares de poda (quanto mais próximo de 0, maior a pontuação de probabilidade de paráfrase), limitando a quantidade de paráfrases de acordo com a probabilidade obtida por cada par. Porém, a medida que a nota sobe, a cobertura cai (acima no gráfico).

Com esses resultados, os autores acreditam que as bases PPDB serão úteis em diversas aplicações de PLN. Em trabalhos futuros, pretendem obter melhores

pontuações para as paráfrases por meio da incorporação de novas fontes de informação e do refinamento dos dados, para tratar ambiguidades e peculiaridades de determinados idiomas pivô e aumentar e melhorar constantemente o conteúdo.

O trabalho de Ganitkevitch et alli (2013) tem uma relação muito forte com o trabalho apresentado nesta dissertação, pois ambos empregam o mesmo método de extração de paráfrases através de corpus paralelo bilíngue utilizando idioma pivô. Além disso, o objetivo de ambos os projetos é a criação de repositórios de paráfrases nos idiomas nativos e ambos investigam o uso de informação de contexto juntamente com as probabilidades calculadas com base em alinhamento.

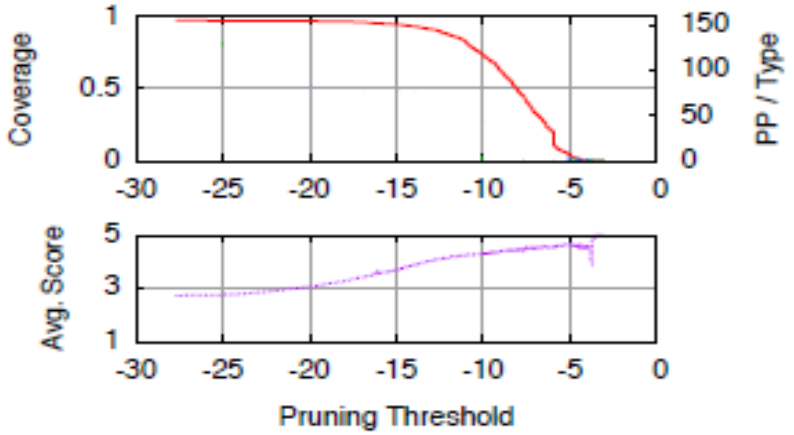


Figura 2.3 - Cobertura considerando diferentes limiares de poda (acima) e pontuação média considerando diferentes limiares de poda (abaixo). Fonte: (GANITKEVITCH et al., 2013).

2.2.4 Szpektor et alli (2004)

No trabalho de Szpektor et alli (2004) é apresentado um algoritmo não supervisionado para a extração de relações semânticas, entre elas a relação de paráfrases, em nível de sintagmas, a partir da *web*. O algoritmo utiliza como entrada um léxico de verbos e busca na *web*, para cada verbo do léxico, modelos de vinculação sintática, ou seja, *templates* que definem a função que cada verbo assume na oração (como sujeito, objeto direto, etc.).

Para isso, o método se baseia nos elementos lexicais em comum nas paráfrases, chamados de âncoras. Âncoras são elementos lexicais que ajudam a identificar o contexto de uma sentença, como sinais de pontuação, nome próprio, etc. A partir das âncoras, torna-se possível criar modelos que reconheçam diferentes sentenças, com diferentes estruturas linguísticas, como sentenças equivalentes.

O processamento do método proposto está dividido em dois algoritmos: o ASE (responsável pela extração do conjunto de âncoras) e o TE (responsável pela extração dos *templates*). O ASE é dividido em quatro etapas, assim como descrito na Tabela 2.3.

Por exemplo, considerando-se como entrada para o algoritmo ASE o verbo "*prevent*" (P), no passo 1 é criada um *template* (Tp) para (P), como "*X subj prevent obj Y*". As lacunas (*slots*) X e Y permitem corresponder Tp com outras possibilidades. No passo 2, a partir do Tp "*X subj prevent obj Y*", constrói-se um corpus S a partir da *web* com sentenças que contenham P e apresentem a mesma estrutura sintática de Tp. No passo 3 são extraídos conjuntos de candidatos a âncora, por exemplo, para a sentença no corpus S "*antibiotics in pregnancy prevent miscarriage*", foi extraída a âncora {*antibiotics subj, miscarriage obj*}. E finalmente, no passo 4, candidatos inadequados são filtrados por sua frequência absoluta na *web* e pela probabilidade condicional do pivô, calculada com base na divisão entre a frequência na *web* da intersecção do pivô *P* com o conjunto de âncoras *c* pela frequência na *web* do conjunto de âncoras *c*.

Tabela 2.3 - Algoritmo ASE. Fonte: (SZPEKTOR et al., 2004, p. 3).

Para cada verbo de entrada:
1 - Criar um *template* de pivô (Tp)
2 - Construir um corpus amostra (S) para Tp
(a) Obter uma amostra inicial da Web
(b) Identificar sentenças associadas a Tp
(c) Estender S usando as sentenças associadas
3 - Extrair candidatos a conjunto de âncoras de S
(a) Extrair âncoras de slots[21]
(b) Extrair âncoras de contexto
4 - Filtrar os conjuntos de candidatos a âncora:
(a) por frequência absoluta
(b) pela probabilidade condicional do pivô

O segundo algoritmo aplicado por Szpektor et alli (2004), o TE, recebe como entrada uma lista de conjuntos âncora criados pelo algoritmo ASE, para cada pivô. Sua execução está dividida em três fases: (1) aquisição de amostra de corpus da *web*; (2) extração do máximo possível de *templates* deste corpus; e (3) classificação dos *templates* extraídos.

[21] O terno "slot", neste caso, refere-se a uma lacuna em uma sentença. Por exemplo, no template "X prevent Y", X e Y são slots.

A aquisição da amostra de corpus via *web* é feita a partir de cada conjunto de âncoras de entrada. Em seguida, os *templates* são extraídos a partir das sentenças desse corpus por meio da substituição das âncoras por variáveis e da aplicação de um algoritmo de aprendizagem estrutural chamado *General Structure Learning* (GSL). No exemplo citado por Szpektor et alli (2004), o par {*aspirin, heart attack*} encontrado na sentença *"Aspirin stops heart attack?"* é substituído, gerando o *template "X stops Y"*. Dessa forma, para cada par de âncoras de um determinado grupo, essas âncoras são unificadas e atribui-se a cada uma a mesma variável (ou seja, "*aspirin"* foi atribuída a X e "*heart attack"*, a Y). Essas sentenças modificadas são, então, processadas pelo algoritmo GSL que extrai *templates* em duas principais etapas: (1) constrói uma representação compacta de grafos de todos os dados extraídos da análise da sentença S e (2) extrai *templates* a partir dessa representação. Na última etapa do processo, o TE remove os *templates* correspondidos por apenas uma âncora. Os *templates* restantes são classificados levando-se em consideração o conjunto de ancoragem com que cada *template* apareceu seguido do número de sentenças em que foi encontrado.

Na avaliação, os algoritmos foram executados em um léxico de verbos e os resultados avaliados por juízes humanos. Foi utilizado um conjunto de 53 verbos selecionados aleatoriamente entre os 1000 verbos mais frequentes de um subconjunto do corpus Reuters (ROSE et al., 2002). Para os 53 verbos, foram gerados 752 *templates*. Os juízes avaliaram os resultados classificando os pares (pivô, *template*) em uma das três classes: (1) correto, se existisse uma relação entre o *template* e o pivô, dentro de um contexto aceitável; (2) incorreto, se não houvesse um contexto razoável para esse vínculo; e (3) sem avaliação, em casos em que os juízes não conseguiram chegar a uma conclusão definitiva.

Os resultados atingiram precisão de 44,15% em média e um rendimento médio de 5,5 *templates* por verbo avaliado. A concordância obtida entre os juízes, dada pelo Kappa (CARLETTA, 1996), foi 0,55 para o Juiz1 e o Juiz2, 0,63 para o Juiz1 e o Juiz3 e 0,57 para o Juiz2 e o Juiz3. Os resultados são apresentados na Tabela 2.4.

Segundo os autores, os resultados são animadores, já que o sistema extraiu uma boa quantidade de *templates* e mostrou capacidade para descobrir relações semânticas complexas, obtendo também *templates* corretos para 46 dos 53 verbos

avaliados, relativos à cobertura global. No entanto, os resultados mostram que o método ainda pode ser melhorado, dependendo da amplitude da busca na *web*.

Tabela 2.4 - Avaliação do método de Szpektor et alli (2004). Fonte: Adaptado de Szpektor et alli (2004).

	Templates corretos	**Templates incorretos**	**Templates sem avaliação**
Juiz1	283(37,63%)	467(62,10%)	2(0,002%)
Juiz2	313(41,62%)	439(58,37%)	0(0%)
Juiz3	295(39,22%)	441(58,64%)	16(0,02%)

O método de Szpektor et alli (2004), assim como a proposta apresentada neste documento, visa o aprendizado a partir da *web*. Contudo, ele é mais abrangente uma vez que busca relações semânticas abertas e não somente paráfrases, como é o caso desta proposta.

2.2.5 Pang et alli (2003)

No trabalho de Pang et alli (2003) é descrito um modelo baseado no alinhamento de árvores sintáticas de sentenças paralelas monolíngues (em inglês), que tem como objetivo extrair paráfrases lexicais e sintáticas em nível de palavras e de sintagmas. As paráfrases lexicais, como descrito na seção 2.1, apresentam significado semelhante, como ocorre em {*conflict, fight*}, {*research, stud}*. As paráfrases sintáticas, por sua vez, são estabelecidas, quando existe uma mudança na estrutura sintática, como em {*last month's conflict, the fight of last month*}. Além disso, o algoritmo também gera novas sentenças que parafraseiam as sentenças de entrada. As paráfrases geradas servem de referência para a avaliação automática de traduções produzidas automaticamente.

O algoritmo proposto, reproduzido na Figura 2.4, é composto por três etapas principais: (i) construção de uma floresta sintática, a partir das árvores sintáticas das sentenças de entrada (passos de 1 a 4 do algoritmo); (ii) transformação da floresta sintática em um autômato de estados finitos (passo 6 do algoritmo) e (iii) compactação do autômato para a geração de novas sentenças (passo 7 do algoritmo).

1. ParseForest = ϵ
2. foreach $s \in SentenceGroup$
3. t = parseTree(s);
4. ParseForest = Merge(ParseForest, t);
5. endfor
6. Extract FSA from ParseForest;
7. Squeeze FSA;

Figura 2.4 - Algoritmo de alinhamento. Fonte: (PANG et al., 2003, p. 3).

Dado um conjunto de sentenças paralelas de entrada, cada sentença é processada por um *parser*, que gera as árvores sintáticas correspondentes (uma para cada sentença). Em seguida, o algoritmo constrói uma floresta sintática, a partir do alinhamento de pares de árvores sintáticas. O alinhamento entre duas árvores sintáticas consiste em identificar e unir os nós que representam informações equivalentes. Esse processo é feito de maneira incremental, ou seja, em um primeiro momento um par de árvores é alinhado e as árvores são unidas, originando a floresta sintática. No momento seguinte, a próxima árvore sintática é alinhada à floresta e unida a ela e, assim, sucessivamente até se obter uma floresta sintática com a fusão de todas as árvores. O alinhamento é realizado de forma *top-down,* ou seja, partindo do nó raiz até os nós-folha. Por se tratar de sentenças paralelas, os autores supõem que os nós pertencentes a uma mesma classe gramatical são paráfrases. Nenhum tipo de conhecimento semântico é usado nesse processo. Assim, são alinhados apenas os nós que apresentam unidades lexicais idênticas ou que têm a mesma classe gramatical.

Para ilustração, a Figura 2.5 mostra um exemplo de floresta sintática (*Parse Forest*, na figura) obtida a partir do alinhamento de duas árvores sintáticas (*Tree* 1 e *Tree 2*, na figura) correspondentes às sentença "*12 persons were killed.*" e "*Twelve people died*.", respectivamente. Como se pode notar na floresta, os nós "*12*" e "*persons*" da primeira árvore (Tree 1) foram alinhados aos nós "*twelve*" e "*people*" da segunda árvore (Tree 2), respectivamente, pois pertencem às mesmas classes gramaticais, isto é, CD (cardinal) e NN (substantivo).

Após o alinhamento (união das árvores), a floresta é mapeada em um autômato finito. O mapeamento é feito percorrendo-se a floresta a partir da raiz até as folhas, criando caminhos alternativos para cada nó alinhado. Em seguida, o autômato passa por um processo de compactação, no qual duas ou mais arestas

com o mesmo nome e que tenham o mesmo nó origem e o mesmo nó destino são transformadas em uma única aresta. Cada caminho no autômato, iniciando no nó *BEG* e terminando no nó *END,* representa uma possível paráfrase das sentenças de entrada.

Com esse método, os autores esperam solucionar dois problemas: o problema da representação de paráfrases, transformando sua representação em *strings* (como "*12 people were killed*") ou em padrões (como "*CD NN AUX VP*"), e o problema da indução de paráfrases, gerando novas paráfrases a partir de um conjunto de entrada. Segundo Pang et alli (2003), a representação de paráfrases baseada em autômatos de estados finitos possibilita codificar um grande número de paráfrases. Considere o exemplo da Figura 2.5, o autômato gerado através da união das duas sentenças "*12 persons were killed.*" e "*Twelve people died*." permite o reconhecimento de 6 novas sentenças: (1) "*12 persons died.*", (2) "*12 people were killed.*", (3) "*12 people died.*", (4) "*Twelve people were killed*.", (5) "*Twelve persons died*." e (6) "*Twelve persons were killed*."). Além disso, essa representação também permite a aplicação de algoritmos que derivam automaticamente tais representações de entradas (PANG et al., 2003).

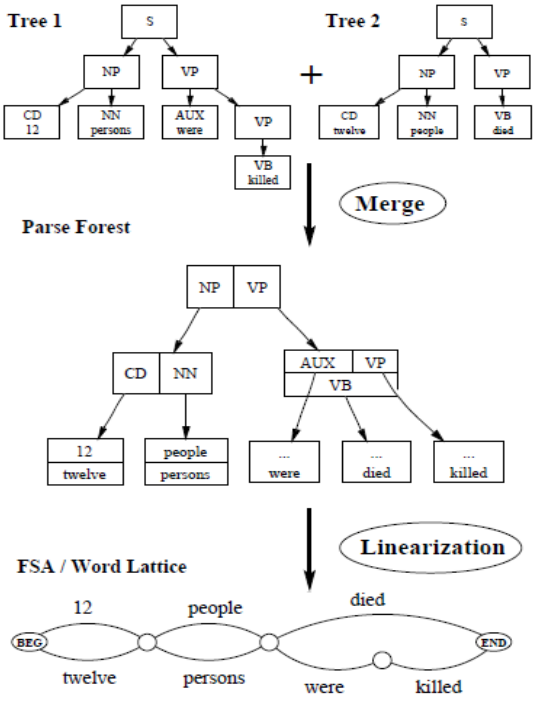


Figura 2.5 - União top-down das árvores e extração do autômato (FSA). Fonte: (PANG et al., 2003, p. 3).

O corpus utilizado pelos autores para a avaliação do método proposto, denominado Multiple-Translation Chinese Corpus (MTC), é composto por 105 artigos jornalísticos, que foram traduzidos do chinês para o inglês por 11 agências de notícias distintas, totalizando, assim, 11 traduções diferentes para cada sentença do corpus (899 sentenças no total).

Para avaliar as paráfrases geradas pelo método, 300 pares de paráfrases foram selecionados aleatoriamente dos autômatos produzidos pelo sistema, e foram comparados com outros 300 pares de paráfrases, também selecionados aleatoriamente da lista de paráfrases resultante do método proposto por Barzilay e McKeown (2001). Esses 600 pares de paráfrases foram apresentados aleatoriamente a 4 juízes humanos, cada um deles instruído a avaliar 150 pares de paráfrases, sendo 75 pares do sistema proposto por Pang et alli (2003) e 75 produzidos pelo método de Barzilay e McKeown (2001). Três opções de avaliação foram sugeridas: “correta”, em casos que ocorrem paráfrases perfeitas; “parcialmente corretas”, em casos em que existe uma sobreposição parcial entre os significados, como no exemplo utilizado por Pang et alli (2003), (*set*, *aid package*), onde o correto seria (*saving set*, *aid package*); e “incorreta”, em casos sem sentido. Os resultados obtidos podem ser vistos na Tabela 2.5.

Tabela 2.5 - Comparativo entre os métodos de Pang et alli (2003) e Barzilay e Mckeown (2001). Fonte: (PANG et al., 2003, p. 5).

	Métodos Avaliados	**Correta**	**Parcialmente Correta**	**Incorreta**
Juiz 1	Pang et alli, 2003	85%	12%	3%
	Barzilay e McKeown, 2001	68%	13%	19%
Juiz 2	Pang et alli, 2003	80%	13%	7%
	Barzilay e McKeown, 2001	63%	13%	24%
Juiz 3	Pang et alli, 2003	81%	5%	13%
	Barzilay e McKeown, 2001	68%	3%	29%
Juiz 4	Pang et alli, 2003	77%	17%	5%
	Barzilay e McKeown, 2001	68%	16%	16%
Média	Pang et alli, 2003	81%	12%	7%
	Barzilay e McKeown, 2001	66%	11%	22%

O algoritmo proposto por Pang et alli (2003), além de apresentar bons resultados, conforme dados da Tabela 2.5, é útil em vários contextos, como na indução de paráfrases lexicais e estruturais, na geração de representações

semanticamente equivalentes e também para estimar a qualidade de sistemas de tradução automática.

2.2.6 Seno e Nunes (2009)

No trabalho de Seno e Nunes (2009), as autoras buscam o reconhecimento de informações comuns em um grupo de sentenças relacionadas, em português, através do reconhecimento de paráfrases em nível de palavras e sintagmas. O objetivo final do método é produzir, através da fusão de sentenças similares, uma nova sentença que contenha as informações comuns a este grupo. Assim como Seno e Nunes (2009), a proposta apresentada neste documento visou o reconhecimento de paráfrases, porém sem perseguir o objetivo maior de fundir sentenças.

Um alinhador de conceitos foi proposto por Seno e Nunes (2009) para o reconhecimento de informações comuns com base no alinhamento de árvores de dependência sintática[22]. A partir do alinhamento de duas ou mais árvores de dependência sintática que representam cada sentença comparável de um conjunto, é construída uma floresta (união de informações comuns de cada sentença). Essa floresta é usada, por um módulo de fusão e linearização, para gerar todas as sentenças possíveis a partir da floresta. Ou seja, a estratégia aplicada aqui é semelhante àquela aplicada em Pang et alli (2003), porém enquanto Pang et alli (2003) alinham estruturas sintáticas de sentenças paralelas, Seno e Nunes (2009) alinham estruturas de dependência sintática de sentenças comparáveis.

As árvores de dependências sintática utilizadas como entrada para o alinhador de informações comuns são obtidas com o *parser* do português Palavras (BICK, 2000) . Durante o alinhamento, o modelo faz uso de um *thesaurus*, o TeP (Thesaurus Eletrônico do Português do Brasil) (MAZIERO et al., 2008), que contempla conjuntos de sinônimos de verbos, substantivos, adjetivos e advérbios, e de uma lista de palavras de classe fechada, como artigos e conjunções, do português (ou *stoplist)*, que auxilia na identificação de palavras irrelevantes para o alinhamento.

[22] Árvore de dependência sintática é uma árvore formada pela análise sintática de sentenças, em termos gramaticais, com a indicação da dependência entre os elementos . Por exemplo, para a sentença “A garota trabalha”, a árvore de dependência sintática exibiria o que é o artigo, o que é o sujeito e o que é o complemento da sentença. (SILVA, 2013).

O alinhamento das sentenças comparáveis é iniciado a partir de um grupo de sentenças (contendo no mínimo duas sentenças), onde o algoritmo identifica, entre as duas primeiras sentenças do grupo, todos os alinhamentos possíveis. As duas sentenças alinhadas são unidas em uma estrutura de dependência sintática dando origem à floresta, onde também serão unidas as outras sentenças pertencentes ao grupo, conforme alinhadas à floresta. O resultado desse processo de alinhamento é, portanto, uma estrutura de dependência sintática contendo todas as sentenças de um grupo, e suas intersecções. Ainda no que diz respeito aos alinhamentos, o método empregado difere dos alinhamentos em outras tarefas do PLN, onde algumas informações estão presentes em apenas uma das sentenças, além do que, palavras pertencentes às classes gramaticais fechadas, como os artigos, por exemplo, são alinhadas apenas na ocorrência de paráfrases multipalavras.

Mais especificamente, dadas duas sentenças de entrada, o algoritmo procura, para cada palavra pertencente à primeira sentença, por possíveis candidatos ao alinhamento na segunda sentença. Para tanto, o método usa como âncora palavras com o mesmo lema ou sinônimas e que tenham o mesmo *PoS* (*Part-of-Speech*). Em seguida, ele calcula, para cada palavra candidata, a sua probabilidade de alinhamento, sendo 0,5 a probabilidade mínima para o alinhamento. Esse alinhamento inicial gerado para as duas primeiras sentenças é, então, iterativamente estendido processando e inserindo uma nova sentença na floresta a cada iteração. Assim, para cada palavra da nova sentença são procuradas candidatas ao alinhamento na floresta. Para cada candidato encontrado, é calculada sua probabilidade de alinhamento.

O cálculo da probabilidade de alinhamento considera: (i) a similaridade semântica entre palavras e sintagmas, (ii) a similaridade sintática (ou seja, se ambos são, por exemplo, sujeito ou objeto direto) e (iii) a similaridade entre os dependentes sintáticos (por exemplo, se ambos são sujeitos de verbos equivalentes). Para a similaridade semântica, além do *thesaurus* o modelo utiliza um conjunto de 27 regras de parafraseamento, obtidas manualmente a partir de estudos do corpus. Nos casos em que tanto a palavra candidata quanto a palavra fonte têm a mesma função sintática, a probabilidade de alinhamento é acrescida em 0,3. Durante o alinhamento dos verbos, o algoritmo verifica se os verbos correspondentes são sinônimos ou paráfrases, adicionando 0,3 na probabilidade de alinhamento em caso positivo. No caso de palavras idênticas, o algoritmo atribui probabilidade 1, e para sinônimos e

cognatos, 0,5. No caso de uma palavra candidata obter pontuação maior ou igual a 0,5, ela é então alinhada à palavra fonte, finalizando a busca por novos candidatos.

O corpus utilizado pelas autoras é um corpus comparável em português construído a partir de 50 coleções de textos jornalísticos provenientes de várias agências de notícias da *web*. Como se trata de um corpus comparável, os documentos de uma mesma coleção se referem a um mesmo assunto. No total, o corpus possui 71 documentos com 1153 sentenças.

A partir do corpus, 30 pares de sentenças comparáveis foram selecionados para formulação de regras de parafraseamento que permitem identificar, em sentenças comparáveis, sequências de palavras distintas com o mesmo significado. A partir desses 30 pares, 81 paráfrases foram identificadas manualmente, considerando que um item pode ser substituído por essas paráfrases sem causar mudanças significativas no contexto. Das ocorrências identificadas, 26% dos casos são paráfrases lexicais (entre palavras simples) e o restante (74%) são paráfrases multipalavras. Para as paráfrases multipalavras foram criadas 27 regras de parafraseamento. Algumas destas regras podem ser vistas na Tabela 2.6, onde ADJ representa adjetivo, ART: artigo, ADV: advérbio, N: substantivo, V: verbo, PRP: preposição, PROP: nome próprio, ?: significa nenhuma ou uma ocorrência, I: indica alternativa (operador ou) e os números: unidades lexicais similares. Por exemplo, dois segmentos, S1 e S2, são considerados paráfrases, segundo a regra R3, por exemplo, se S1 iniciar com um substantivo, seguido de um nome próprio e S2 iniciar com um substantivo, acompanhado de um adjetivo e terminando com um nome próprio. Para as paráfrases lexicais não há regras.

Tabela 2.6 - Exemplos de regras de parafraseamento. Fonte: (SENO e NUNES, 2009, p. 7).

R1. N1 ADJ ; N1 PROP? PRP ART? PROP
R2. N1 ; N1 ADJ
R3. N PROP1 ; N ADJ PROP1
R4. ADV? V PRP V1 ; ADV? V1
R5. N PRP ART? (PROP1|N1); (PROP|N) PRP ART? (PROP1|N1) PRP? ART? (PROP|N)?

Para servir de referência (*gold standard*) na avaliação do sistema, 20 pares de sentenças (diferentes dos pares usados para criação das regras) foram extraídos do corpus e alinhados manualmente por dois anotadores. A avaliação foi realizada com

base nas medidas Precisão, Cobertura e Medida-f[23] e os resultados do método proposto foram comparados a outros dois sistemas *baseline*. O primeiro *baseline,* chamado *Baseline* 1, identifica apenas segmentos idênticos (mesmo lema) ou sinônimos. O segundo, chamado *Baseline* 2, é uma extensão do primeiro, que inclui também os traços de dependência sintática, ou seja, considerando alinhamentos lexicalmente similares, o mesmo PoS e os mesmos traços de dependência. Os resultados podem ser vistos na Tabela 2.7.

Tabela 2.7 - Resultados do alinhamento automático. Fonte: (SENO e NUNES, 2009, p. 14).

Sistema	Precisão	Cobertura	Medida-f
Baseline 1	0,81	0,76	0,78
Baseline 2	0,80	0,75	0,78
Alinhador Proposto	0,87	0,83	0,85

Conforme mostram os resultados da Tabela 2.7, o sistema proposto obteve um ganho de cerca de 7 pontos percentuais em relação aos *baselines*. Em se tratando apenas do alinhamento de paráfrases (ou seja, desconsiderando-se casos de casamento idêntico), o sistema obteve um desempenho ainda melhor em relação aos *baselines*, sendo de até 40 pontos percentuais para a Medida-F. Os resultados são mostrados na Tabela 2.8.

Tabela 2.8 - Resultados do alinhamento automático considerando apenas os casos de paráfrases. Fonte: (SENO e NUNES, 2009, p. 15).

Sistema	Precisão	Cobertura	Medida-f
Baseline 1	0,55	0,14	0,23
Baseline 2	0,53	0,24	0,33
Alinhador Proposto	0,69	0,60	0,64

Como pôde ser visto o sistema proposto por Seno e Nunes (2009) apresentou bons resultados, principalmente quando levado em consideração apenas o alinhamento de paráfrases (excluindo os casos de segmentos idênticos), tarefa equivalente à perseguida neste projeto.

2.2.7 Aziz e Specia (2013)

Segundo Bannard e Callison-Burch (2005), é possível extrair paráfrases através de alinhamentos de sentenças em um corpus paralelo bilíngue. Como já

[23] A Medida-f é uma média harmônica entre a cobertura e a precisão.

descrito anteriormente (veja seção 2.2.2), essa abordagem consiste na extração de paráfrases através do alinhamento entre sentenças paralelas (ou seja, o alinhamento da sentença no idioma alvo com sua tradução no idioma pivô). Nesta abordagem, é provável que ocorram vários alinhamentos (candidatos) em cada sentença alvo, no entanto, as ambiguidades inerentes à língua pivô podem acarretar em paráfrases inadequadas.

Por esse motivo, Aziz e Specia (2013) propuseram uma nova formulação para o método de Bannard e Callison-Burch (2005), que pode ser utilizada para reconhecimento tanto de paráfrases em nível de palavras quanto em paráfrases em nível de sintagmas contendo até sete palavras. Esse método apresenta uma mudança que é tratada por Aziz e Specia (2013) como anotação "*quasi-sense*" (semi-significado), restringindo o sentido da informação. Em outras palavras, as traduções obtidas com os alinhamentos de uma determinada sentença alvo são utilizadas para restringir as suas interpretações possíveis, ou seja, cada paráfrase candidata se limitaria a uma dessas possíveis interpretações. Para isso, esse método necessita não só de um corpus paralelo bilíngue, referente aos idiomas alvo e pivô, mas também de outros corpora paralelo, entre as línguas pivô e a linguagem utilizada para atribuir as interpretações da sentença, definida por *q*.

Para tratar do "*quasi-sense*", é adicionada uma etiqueta à sentença fonte, que consiste em uma sentença na língua pivô, ou seja, uma tradução válida da sentença alvo. A Equação (2), que é uma adaptação do modelo de Bannard e Callison-Burch (2005), define o alinhamento de *e2* (candidato a paráfrase de *e1*), com *e1* (a ser parafraseado) através do alinhamento com *f* (tradução de *e1* no idioma pivô), que também produz etiquetas de sentido *q,* produzidas através do alinhamento de *e1* com *f* e do alinhamento de *f* com a sentença correspondente no idioma usado para gerar as etiquetas de sentido*. e2* e *q* são independentes de *f*.

$$p(e2, q|f) \overset{\mathrm{e2}}{=} \coprod^{\mathrm{q}} |\mathrm{f}\, \mathrm{p}(\mathrm{e2}|\mathrm{f}) p(q|f) \tag{2}$$

$$p(e2|e1, q) = \frac{1}{Z} \sum_{f \in F} \quad p(e2|f) p(q|f) p(f|e1) \tag{3}$$

Por exemplo, considerando como entrada a palavra "*casa*" (*e1 – em português*), a Equação (2) deverá encontrar o candidato "*residência*" (*e2 – em português*) através do alinhamento com "*residenz*" (*f – em alemão*) e encontrar etiquetas de sentido *q*, como "*dwelling*" e "*marry*" (ambas em inglês, nesse caso, idioma utilizado para gerar as etiquetas de sentido). Essa comparação exige um

corpus trilíngue, pois o idioma das etiquetas de sentido deve ser diferente dos idiomas alvo e pivô.

A Equação (3) mostra como são calculadas as probabilidades, através do pivô, considerando a probabilidade de alinhamento que *e1* obteve em relação à *f* (considerando todos os alinhamentos) e a probabilidade que *e1* assume (através do sentido da etiqueta *q*) em relação a todos os alinhamentos.

Como a proposta de Aziz e Specia (2013) utiliza vários idiomas como pivô, foi utilizada a coleção de dados Europarl (KOEHN, 2005), visando parafrasear sentenças em espanhol (idioma alvo), e suas correspondentes em inglês foram utilizadas para etiquetar as interpretações. Além destes idiomas, outros nove foram utilizados como pivô: alemão (de), holandês (nl), dinamarquês (da), sueco (sv), finlandês (fi), francês (fr), italiano (it), português (pt) e grego (el). As sentenças do corpus paralelo são, primeiramente, alinhadas em nível de palavras utilizando GIZA++ tanto nas direções fonte-alvo quanto alvo-fonte.

As probabilidades para os significados que a palavra pode assumir são calculadas como mostrado na Equação (3), utilizando frequências relativas, ou seja, para cada sentença em espanhol, as paráfrases candidatas são encontradas e classificadas de acordo com o sentido que assumem (etiquetas de sentido, escritas em inglês) através de nove corpora bilíngues utilizados como pivô. Pela separação dos candidatos em sentidos, é possível calcular a probabilidade de cada sentido, através da média das frequências em que seus candidatos são alinhados.

Foi criado também um conjunto de testes contento vários sintagmas polissêmicos. Foram utilizadas sintagmas em espanhol para prover casos de ambiguidade, selecionadas a partir da versão espanhola da WordNet. Dos 50 sintagmas selecionados (contendo pelo menos uma palavra), 40 delos tinham pelo menos duas interpretações diferentes. É essencial relatar que, em sintagmas com mais de uma interpretação, para cada interpretação foi utilizada uma tradução diferente como etiqueta, de maneira a evitar alinhamentos inconsistentes.

Para fins de avaliação, o modelo de Aziz e Specia (2013) foi comparado com duas variações do modelo apresentado por Bannard e Callison-Burch (2005). Para cada um destes modelos foram parafraseadas 258 amostras do conjunto de testes. Apenas as três melhores paráfrases referentes a cada modelo foram selecionadas manualmente e avaliadas.

Os julgamentos foram baseados na forma como Bannard e Callison-Burch (2005) avaliaram seu método, ou seja, os sintagmas foram apresentados aos julgadores inseridos em sua sentença original e, então, substituídos por seus respectivos candidatos à paráfrase. Os juízes avaliaram a qualidade destas substituições em dois aspectos: (i) se foi mantido o mesmo significado da sentença original e (ii) se a gramaticalidade da sentença não foi afetada. Apenas os candidatos que preservavam os dois aspectos foram considerados corretos. Para cada sintagma a ser parafraseado, foram selecionadas as três melhores paráfrases de cada método comparado, considerando dois cenários de avaliação:

(1) traduções baseadas em um *gold-standard* – formado pelo conjunto das etiquetas de interpretação, como as utilizadas para palavra espanhola "forma": *means/way of doing/achieving something*, *shape*, *type or group sharing common traits*, em inglês, encontradas no Europarl;

(2) traduções baseadas em Tradução Automática Estatística – criadas com a utilização de *Moses*[24] (KOEHN et al., 2007) e o conjunto de sentenças espanhol-inglês utilizados no trabalho, exceto as sentenças do conjunto de teste.

Os julgamentos foram realizados por 7 falantes nativos do espanhol, julgando um total de 5.110 sentenças. A medida de concordância Kappa (CARLETTA, 1996) foi de 0,54 ± 0,15 para julgamentos quanto ao significado, 0,63 ± 0,16 para julgamentos quanto a questões gramaticais e 0,62 ± 0,20 para exatidão. Além das avaliações quanto à manutenção do significado e da gramaticalidade das sentenças resultantes do parafraseamento, Precisão e Cobertura também foram calculadas para os três melhores candidatos de cada modelo. Os resultados são apresentados na Tabela 2.10 onde *Top* representa os três melhores candidatos para cada exemplo, *BCB* representa o modelo proposto por Bannard e Callison-Burch (2005), *BCB-Mod* representa uma extensão do modelo de Bannard e Callison-Burch (2005), usando etiquetas de interpretação (em inglês) e *AS* representa o modelo proposto por Aziz e Specia (2013).

Como pôde ser visto na Tabela 2.9, o método de Aziz e Specia (2013) apresenta melhores resultados, tanto em Precisão quanto em Cobertura, quando comparado com o modelo de Bannard e Callison-Burch (2005), mesmo quando este

[24] Moses é um conjunto de ferramentas de código aberto aplicáveis a várias tarefas do processo de tradução, como a geração de modelos de língua e tradução usados na geração de tradutores automáticos estatísticos.

último foi modificado. Isso se deve aos 9 idiomas pivôs utilizados por Aziz e Specia (2013), o que garante acessibilidade a candidatos que não seriam encontrados por Bannard e Callison-Burch (2005). Além disso, o método de Aziz e Specia (2013) apresenta a vantagem sobre o método de Bannard e Callison-Burch (2005) por reconhecer paráfrases que o método original (Bannard e Callison-Burch (2005)) não reconhece, devido ao sentido que cada sintagma pode assumir.

Tabela 2.9 - Resultados do método de Aziz e Specia (2013). Fonte: Adaptado de Aziz e Specia (2013).

Método	Top	Significado	Gramaticalidade	Exatidão		
		Medida F	Medida F	Precisão	Cobertura	Medida F
BCB	1	32	28	25	25	25
BCB-Mod	1	61	38	34	28	30
AS	1	62	55	59	42	49
BCB	2	41	37	33	33	33
BCB-Mod	2	68	44	40	33	36
AS	2	71	64	66	47	55
BCB	3	46	42	37	37	37
CCB-Mod	3	71	47	45	36	40
AS	3	74	67	71	50	59

2.2.8 Van Der Plas e Tiedemann (2006)

O trabalho de Van Der Plas e Tiedemann (2006) visa o reconhecimento de sinônimos em holandês, que são definidos pelos autores como palavras simples/isoladas que apresentam a mesma ideia. São apresentados dois métodos: o primeiro trata da extração de sinônimos através de similaridade distribucional baseada no alinhamento lexical (DSBWA) e o segundo através de similaridade distribucional baseada em relações sintáticas (DSBSR). Um dos objetivos é comparar os resultados dos métodos apresentados.

O método DSBWA utiliza corpus paralelo multilíngue alinhado com o GIZA++ (OCH e NEY, 2003). A Identificação dos sinônimos feita por Van Der Plas e Tiedemann (2006) é similar à forma como é feita nos trabalhos de Bannard e Callison-Burch (2005) e neste, identificados através de similaridade de traduções. Ou seja, se duas palavras têm traduções semelhantes ou seus contextos têm traduções semelhantes, então são consideradas sinônimos. O corpus utilizado no método DSBWA é composto por onze idiomas diferentes.

O outro método apresentado, o DSBSR, utiliza corpus paralelo monolíngue. A ideia por traz desse método é que sinônimos compartilham contextos similares. Em

outras palavras, a hipótese dos autores é que duas palavras A e B que compartilham os mesmos contextos são semanticamente relacionadas.

No cálculo da similaridade distribucional, as palavras do contexto de uma palavra investigada são utilizadas como *features* e são representadas por meio de um vetor de características, que contém as frequências de cada *feature*, de acordo com os vários contextos onde a palavra é encontrada. Para a identificação de possíveis sinônimos, os vetores de duas palavras quaisquer são comparados entre si para se determinar a sua similaridade.

Na comparação entre os vetores, é utilizada a frequência ponderada, calculada por meio da divisão do número de vezes que determinada palavra ocorreu em determinado contexto pela sua frequência (ou seja, número de ocorrências), conforme apresentado na Equação (4).

Van Der Plas e Tiedemann (2006) acreditam que a frequência ponderada traz benefícios para o cálculo de similaridade distribucional, considerando que palavras que aparecem em muitos contextos distintos possam ser ruídos. O cálculo utilizado para a ponderação é baseado em Church et ali (1989), assim como mostrado na Equação (4), onde *W* é a palavra-investigada, *P(W)* é a frequência da palavra, *P(f)* é a frequência do contexto avaliado e *P(W,f)* é a frequência da ocorrência da palavra junto ao contexto avaliado.

$$I(W,f) = log \frac{P(W,f)}{P(W)P(f)} \quad (4)$$

Os resultados obtidos pelos dois métodos foram avaliados automaticamente. Foi criado um modelo de sinônimos (*gold standard*) com os substantivos presentes na EuroWordNet (VOSSEN, 1998). Assim, o *gold standard* foi formado por todos os substantivos em holandês. Dessa forma é possível avaliar os resultados automaticamente comparando os resultados com o *gold standard*. A precisão foi calculada como a porcentagem dos sinônimos candidatos que realmente são sinônimos, ou seja, estão presentes no gold standard. Nesse teste foram utilizados 1000 substantivos com frequência igual ou superior a cinco ocorrências. Já na avaliação da cobertura, muitos sinônimos foram considerados incorretos por conta de não estarem presentes no EuroWordNet, apesar de parecerem corretos na visão dos autores.

O corpus paralelo usado pelo método DSBWA é o Europarl (KOHEN, 2002). Foram utilizadas sentenças alinhadas em 11 idiomas, contendo 1,2 milhão de

palavras em holandês. Foi feita a lematização do corpus antes do alinhamento lexical, a fim de reduzir a escassez de dados. Outro processo executado foi a remoção de alinhamentos que ocorrem apenas uma vez. Os vetores de contexto são preenchidos com os *links* para as palavras que são traduções do contexto em questão, além da frequência de cada alinhamento.

A Tabela 2.10 mostra os resultados de Precisão, Cobertura e Medida-f para os métodos DSBSR e DSBWA (utilizando corpus multilíngue), para 3 iterações.

Tabela 2.10 - Resultados obtidos através da aplicação dos métodos baseados em relações sintáticas e alinhamento lexical. Fonte: Adaptado de Van Der Plas e Tiedemann (2006).

	Iteração 1			**Iteração 2**			**Iteração 3**		
	Prec	Cob	M-f	Prec	Cob	M-f	Prec	Cob	M-f
DSBWA	22,3	5,6	9,0	16,4	7,9	10,7	13,3	9,3	10,9
DSBSR	8,8	2,5	3,9	6,9	4,0	5,1	5,9	5,1	5,5

É possível notar uma grande diferença de desempenho entre o DSBSR e o DSBWA. Em se tratando de recursos, o primeiro leva vantagem, já que, segundo Van Der Plas e Tiedemann (2006), contém mais sentenças, pois não existe tamanha fonte de dados multilíngue. Porém, em se tratando de precisão, deixa a desejar. Já o método DSBWA, apesar de contar com menos recursos, leva grande vantagem em relação ao DSBSR, atingindo números mais significativos, tanto em precisão quanto em cobertura, como sugere a tabela 2.10.

É possível notar semelhança entre o trabalho de Van Der Plas e Tiedemann (2006) e o trabalho aqui apresentado, principalmente em relação ao método DSBWA.

2.2.9 Simohata e Sumita (2002)

O trabalho de Simohata e Sumita (2002) apresenta um método para reconhecimento de sinônimos em corpus paralelo bilíngue inglês–japonês empregado na tradução automática baseada em exemplos (Example-Based Machine Translation - EBMT).

A ideia da EBMT é que, através da tradução de uma sentença de entrada se pode adquirir mais de uma sentença semelhante, assim como acontece no processo feito por humanos. Simohata e Sumita (2002) buscam, utilizando EBMT, a extração

de expressões sinônimas (chamadas de SE) através de comparações entre sentenças sinônimas (chamadas de SS).

As SS são definidas como “sentenças com o mesmo significado básico e diferenças lexicais” (SIMOHATA e SUMITA, 2002). As semelhanças entre duas sentenças é medida de acordo com as diferenças em nível de palavras. Dessa forma, sentenças que tem várias palavras em comum, são consideradas como SS. Podem ser citadas como exemplos de SS: *May I take photos?*; *Can I take pictures?* e *Can I take a photo?.*

Já as SE são formadas por palavras e estruturas diferentes quando comparadas, porém, nesse caso, é levado em consideração o contexto onde as candidatas estão inseridas. Dessa forma, duas palavras são consideradas sinônimas se ambas apresentam o mesmo sentido dentro do contexto. Usando o exemplo anterior, *picture* e *photo* são sinônimos apenas se as duas tiverem o mesmo sentido, que no exemplo é *foto* (ou *retrato*). Em casos onde *pictures* tem o significado de *pintura*, por exemplo, *picture* e *photo* não são consideradas sinônimos. Isso torna possível que ocorram casos em que duas palavras sejam sinônimas em um determinado contexto, porém, individualmente, não apresentem o mesmo significado.

O método utilizado para extração de SE foi desenvolvido para extrair SE em inglês através de traduções no idioma japonês e SE em japonês através de traduções no idioma inglês. O que viabiliza a sua execução é que no corpus utilizado pelos autores, ou seja, corpus paralelo bilíngue alinhado sentencialmente, existem conjuntos de sentenças em japonês, J_n e sentenças em inglês, E_n (onde $_n$ expressa o número de identificação de cada sentença), que são traduções, como por exemplo, J_1 e E_1, e ocorrem também sentenças iguais, como por exemplo, J_5 e J_{29} (representando a quinta e a vigésima nona sentenças do corpus japonês, respectivamente), e E_7 e E_{231} (representando a sétima e a ducentésima trigésima primeira sentença do corpus inglês, respectivamente). Isso permite afirmar que se duas sentenças são iguais, suas respectivas traduções também são, possibilitando o reconhecimento de correspondências entre elas.

Essa característica do corpus utilizado permite também que seja possível agrupar todas as sentenças equivalentes em grupos de SS. Os grupos são criados considerando, além da equivalência entre sentenças citada acima, a estrutura frasal em relação à PoS de cada sentença.

A extração de pares de SE é baseada na forma como as SS são formadas, porém considerando as diferenças de contexto. É baseada em programação dinâmica (*DP-match*), baseado nas diferenças que ocorrem em expressões sinônimas. O DP-match é calculado através da distância entre *strings*, onde a distância é o número mínimo de operações como inserção e exclusão, necessários para mapear sequências de caracteres (KORFHAGE, 1997). Outro fator importante da extração de SE é a atribuição de frequência a cada par. Esse processo é baseado nos grupos de SS, ou seja, através do número de grupos de SS que o par SE aparece. Pares de SE que apresentam uma frequência inferior a 5% ou com apenas uma ocorrência são excluídos.

Em experimento, Simohata e Sumita (2002) demonstram o efeito da utilização de sinônimos em dois sistemas EBMT: uma EBMT que utiliza o método apresentado e uma EBMT que não utiliza o método apresentado (convencional). Essa comparação se dá através de dois critérios: melhoria na cobertura e melhoria na qualidade das traduções.

A qualidade das extrações de sinônimos foi avaliada por falantes nativos de japonês (para os resultados da língua fonte) em inglês (para os resultados da língua alvo). Cada caso foi assinalado como correto ou incorreto. Foram avaliadas 1048 sentenças considerando traduções inglês-japonês e 1094 sentenças considerando traduções japonês-inglês.

O resultado das avaliações mostra que, para traduções japonês-inglês, o método de Simohata e Sumita (2002) apresenta 0,7% a mais de precisão em relação à EBMT convencional (90,6% e 89,9%, respectivamente). Para traduções inglês-japonês, a precisão obtida é exatamente a mesma (97,6%).

O método apresentado por Simohata e Sumita (2002) mostra que, apesar de obter um pequeno aumento na precisão em uma das direções de tradução investigadas (inglês-japonês), apresenta a vantagem de não requerer grande conhecimento linguístico para a extração.

2.2.10 Wu e Zhou (2003)

O Método proposto no trabalho de Wu e Zhou (2003) visa à extração automática de sinônimos através da utilização de três recursos: dicionário monolíngue, corpus monolíngue e corpus paralelo bilíngue. As extrações são feitas

por sistemas que utilizam, individualmente, cada um dos três recursos citados acima e também utilizando os três recursos em conjunto. Segundo Wu e Zhou (2003), a grande vantagem dessa metodologia é que os recursos são complementares, tornando possível atingir melhores resultados em relação à abordagens que utilizam apenas um recurso.

A extração de sinônimos utilizando dicionário monolíngue é feita através da criação de vetores contendo as *features* de cada palavra investigada, usando o dicionário *Online Plain Text Dictionary* (BLONDEL e SENNELART, 2002). Cada vetor é formado por palavras que são utilizadas para definir a palavra investigada e também por palavras cujas definições incluem a palavra investigada. Dessa forma, cada vetor contém as características de cada palavra, tornando possível calcular a similaridade entre duas palavras através da comparação de seus vetores (WU E ZHOU, 2003).

A extração de sinônimos utilizando corpus paralelo bilíngue é feita através de traduções inglês-chinês (219.404 palavras em inglês, com três traduções em média para cada palavra), partindo do pressuposto que duas palavras são sinônimas se suas traduções são semelhantes, através de uma palavra em inglês e suas traduções para o chinês. Assim como no método monolíngue, para cada palavra investigada, é construído um vetor de *features* só que desta vez o vetor contém as traduções da palavra e suas probabilidades de tradução calculadas com base nos resultados do alinhamento de palavras. A semelhança de duas palavras também é estimada por meio das semelhanças de seus vetores.

A extração de sinônimos utilizando corpus monolíngue parte do pressuposto que palavras sinônimas tendem a ter contextos similares. Na aplicação desse método foram utilizados artigos do *Wall Street Journal* (de 1987 a 1992). Palavras que têm relação de dependência com a palavra investigada, por meio de seus contextos no corpus monolíngue são utilizadas para extrair correlações. As relações entre as palavras correlatas são representados em forma de triplas formadas por *palavra1*, *palavra2* (*w1* e *w2*, respectivamente) e pelo tipo de relação existente entre elas (<*w1,Relation Type,w2*>). Por exemplo, a partir da sentença *"I declined the invitation"* são geradas as triplas: <*decline, SUBJ,I*>, <*decline, OBJ,invitation*> e <*invitation,DET,the*>. Nesse caso, a similaridade entre duas palavras é calculada usando uma versão ponderada do coeficiente de *Dice*, que avalia quão semelhante são as sentenças através do número comum de bigramas que contém.

Essas três formas de extração citadas acima são tratadas por Wu e Zhou (2003) como classificações binárias. Assim é possível combinar os resultados de cada método e atribuir pontuação a cada candidata através de médias ponderadas.

Para Wu e Zhou (2003), se dois ou mais dos três métodos apresentados extraírem a mesma palavra como sinônimo de uma palavra investigada, existe uma forte tendência de que a palavra seja realmente um sinônimo da palavra investigada. Isso garante melhora na precisão dos sinônimos extraídos. Além disso, se a pontuação obtida por um candidato for significativa, ele também é selecionado, mesmo se extraída por apenas um método.

As quantidades médias de sinônimos obtidos com a aplicação dos três métodos, separadamente, são mostradas na Tabela 2.11.

Tabela 2.11 - Resultados da aplicação dos métodos de extração de sinônimos. Fonte: Adaptado de Wu e Zhou (2003).

	Categoria	Entradas	Sinônimos (Média)
Dicionário Monolíngue	Substantivo	16963	4.7
	Verbo	5084	7.1
Corpus Bilíngue	Substantivo	26253	10.2
	Verbo	7364	14.8
Corpus Monolíngue	Substantivo	16963	4.6
	Verbo	5084	7.1

A precisão, cobertura e medida-F também foram calculados automaticamente por meio da comparação direta dos sinônimos extraídos com aqueles presentes em dois thesauri: WordNet 1.6[25] e Roget's II[26]. Um conjunto de testes foi construído considerando palavras de frequência alta (mais de 100 ocorrências) com 600 substantivos e 340 verbos, palavras de frequência média (entre 10 e 100 ocorrências) com 2000 substantivos e 1300 verbos e palavras de frequência baixa (menos de 10 ocorrências) com 1000 substantivos e 800 verbos. Os resultados estão expressos na Tabela 2.12.

A combinação dos três métodos de extração de sinônimos traz sem dúvida melhores resultados, tanto em precisão como em cobertura, se comparados os resultados com cada um dos métodos empregados individualmente. O único caso em que o resultado dos métodos combinados fora superado foi na cobertura dos substantivos de frequência alta, onde, para o método de extração de sinônimos utilizando corpus bilíngue, fora obtido 0.209.

[25] Disponível em http://www.cogsci.princeton.edu/~wn/.
[26] Disponível em http://www.bartleby.com/thesauri/.

Tabela 2.12 - Resultados obtidos pela aplicação dos métodos combinados. Fonte: Adaptado de Wu e Zhou (2003).

	Avaliação de substantivos								
	Frequência Alta			Frequência Média			Frequência Baixa		
	Prec	Cob	M-f	Prec	Cob	M-f	Prec	Cob	M-f
Dicionario Monolíngue	0.174	0.140	0.155	0.212	0.137	0.167	0.198	0.119	0.149
Corpus Bilíngue	0.225	0.209	0.217	0.242	0.212	0.226	0.207	0.212	0.209
Corpus Monolíngue	0.118	0.109	0.114	0.117	0.104	0.109	0.099	0.096	0.098
Métodos combinados	0.240	0.201	0.219	0.271	0.220	0.243	0.222	0.232	0.227
	Avaliação de verbos								
	Frequência Alta			Frequência Média			Frequência Baixa		
	Prec	Cob	M-f	Prec	Cob	M-f	Prec	Cob	M-f
Dicionario Monolíngue	0.228	0.243	0.235	0.272	0.233	0.251	0.209	0.216	0.212
Corpus Monolíngue	0.226	0.312	0.262	0.224	0.292	0.253	0.184	0.275	0.220
Corpus Bilíngue	0.143	0.116	0.154	0.162	0.127	0.142	0.128	0.135	0.132
Métodos combinados	0.295	0.323	0.308	0.311	0.304	0.307	0.238	0.302	0.266

2.3 Considerações Finais

Como apresentado neste capítulo, diversos trabalhos foram propostos para o reconhecimento e a extração automáticos de paráfrases e de sinônimos. Tais propostas divergem em termos dos recursos utilizados (corpus/dicionário monolíngue versus corpus paralelo/comparável bilíngue/multilíngue), os quais guiam a escolha pelo método empregado na busca pela similaridade (contexto versus tradução) entre os candidatos.

Para facilitar a comparação, por parte do leitor, dos métodos apresentados na Seção 2.2, a Tabela 2.13 apresenta um resumo das principais características desses trabalhos.

Tabela 2.13 - Resumo das principais características dos trabalhos relacionados apresentados na seção 2.3.

Extração de paráfrases	Tipo de corpus	Tipo de conhecimento utilizado	Granularidade da paráfrase	Idioma
Barzilay e McKeown (2001)	Paralelo monolíngue	Similaridade de contexto	Sintagmas	ingles
Bannard e Callison-Burch (2005)	Paralelo bilíngue	Estatístico (modelos da IBM)	Sintagmas	inglês-alemão
Ganitkevitch et al. (2013)	Paralelo bilíngue	Estatístico	Sintagmas	inglês-alemão
Szpektor et al. (2004)	Web	Estatístico (busca relações de vinculação na web)	Sintagmas	Inglês
Seno e Nunes (2009)	Comparável monolíngue	Similaridade lexical, sintática e semântica (dicionários de sinônimos) por meio do alinhamento de árvores de dependência sintática	Sintagmas	Português
Pang et al. (2003)	Paralelo monolíngue	Similaridade lexical e sintática por meio do alinhamento de árvores sintáticas (sem delimitação de sintagmas)	Sintagmas	inglês-chinês
Extração de sinônimos				
Aziz e Specia (2013)	Paralelo bilíngue	Estatístico (modelos da IBM e desambiguação lexical de sentido)	Sintagmas	espanhol + 8 línguas
Van Der Plas e Tiedemann (2006)	Paralelo multilíngue	Similaridade lexical	Palavras	Holandês + 10 línguas
	Paralelo monolíngue	Similaridade distribucional		Holandês
Simohata e Sumita (2002)	Paralelo bilíngue	Similaridade lexical e sintática baseada em exemplos, por meio de EBMT.	Palavras	inglês-japonês
Wu e Zhou (2003)	Dicionário monolíngue	Similaridade lexical	Palavras	Inglês
	corpus monolíngue	Similaridade de contexto		Inglês
	corpus paralelo bilíngue	Estatístico (através da semelhança entre traduções)		inglês-chinês

Capítulo 3

NEPaL

Este capítulo tem como objetivo descrever todo o planejamento do sistema aprendiz de paráfrases desenvolvido neste projeto, o Never-Ending Paraphrase Learner (NEPaL), desde sua estrutura, passando por uma descrição completa da implementação, do funcionamento de seus módulos, aplicação de técnicas e utilização de ferramentas para seu funcionamento. Devido à utilização de ferramentas externas, essa aplicação só terá pleno funcionamento quando executada sobre a plataforma Linux. Neste projeto, o sistema operacional utilizado é o Ubuntu, versão 14.10. Todo o desenvolvimento do código do NEPaL foi feito na linguagem Java, utilizando a IDE Netbeans.

Esse capítulo está organizado da seguinte forma. Na Seção 3.1 é apresentado o Módulo Coletor; Na Seção 3.2 é apresentado o Módulo Pré-Processador; Na Seção 3.3 é apresentado o Módulo Processador; e Na Seção 3.4 é apresentado o Módulo Promotor.

Retomando, o projeto descrito neste documento teve como objetivo:

"*Verificar se é possível utilizar a estratégia de aprendizado sem-fim e a Internet para aprender, de modo incremental e automático, conhecimento útil para a identificação e a extração de paráfrases."*

Nesse contexto foi desenvolvido o aprendiz de paráfrases deste projeto, o NEPaL. O NEPaL (*Never-Ending Paraphrase Learner*), tem como intuito extrair paráfrases em nível de palavras a partir de um corpus paralelo bilíngue, seguindo o método proposto por Bannard e Callison-Burch (2005). Sua estrutura é baseada no *Never-Ending Bilingual Entity Learner* (NEBEL) proposto por Vieira e Caseli (2013) e seu processamento é dividido em 4 módulos de processamento em *pipeline*: módulo Coletor, módulo Pré-processador, módulo Processador e módulo Promotor. A Figura 3.1 traz uma ilustração dos 4 módulos do NEPaL.

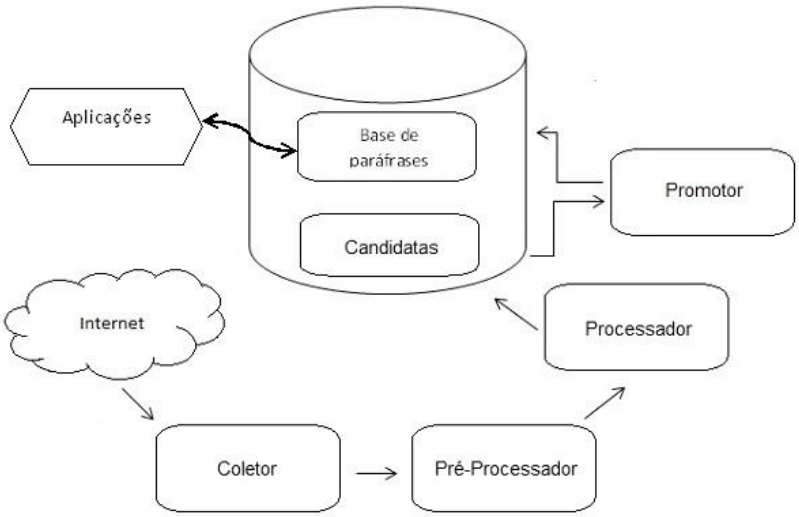


Figura 3.1 - Arquitetura do NEPaL.

Cada um dos módulos da Figura 3.1 executa tarefas sequencialmente, de forma automática, ou seja, sem necessidade de supervisão humana. Assim, os dados resultantes de um módulo são utilizados como ponto de partida no módulo seguinte, conforme sugere a Figura 3.1. Na sequência, (seções 3.1, 3.2, 3.3 e 3.4) apresentam a descrição detalhada de cada módulo.

3.1 Módulo Coletor

O módulo Coletor é responsável por acessar páginas da *web* com o objetivo de adquirir conteúdo textual para a formação do corpus utilizado para o aprendizado de paráfrases. Nesse projeto, optou-se pela construção de um corpus paralelo bilíngue, o mesmo tipo usado no trabalho que inspirou este projeto. O corpus é formado por notícias publicadas na versão internacional do jornal *online* Folha de São Paulo. Esse é um recurso recente, com boa qualidade textual e em constante crescimento, características que viabilizam a criação de um corpus adequado ao aprendizado sem-fim. Nesse caso, para cada artigo (notícia) em inglês publicado na página, sempre existe uma versão original correspondente em português, o que viabiliza o uso do método de Bannard e Callison-Burch (2005) escolhido para o aprendizado de paráfrases no NEPaL.

O módulo Coletor é executado em quatro etapas: (1) captação das *urls* das notícias, (2) captação do conteúdo textual de notícias em inglês, (3) captação dos textos das notícias originais em português (textos paralelos aos captados na etapa 2) e (4) armazenamento (e registro no banco de dados) dos arquivos coletados.

A primeira etapa inicia com o acesso à página da versão internacional da Folha de São Paulo, disponível em www1.folha.uol.com.br/internacional/en/. Nessa página são disponibilizadas notícias diariamente, principalmente sobre fatos ocorridos no cenário nacional. Essas notícias correspondem às versões traduzidas, para o inglês, de notícias originais em português, publicadas anteriormente. Em seguida, são extraídas todas as *urls* dos artigos disponíveis no código fonte da página naquele instante. Não existe um número exato de publicações de artigos disponíveis nem um período exato para cada notícia ficar disponível. A quantidade de *urls* varia, geralmente, entre 25 e 40.

A segunda etapa consiste em acessar todas as *urls* coletadas e, para cada *url*, coletar o conteúdo do artigo propriamente dito, ou seja, desde onde se inicia o texto da matéria até seu término. Não existe um padrão na estrutura do código das páginas da Folha Internacional, por isso, para cada página acessada, sua estrutura é verificada automaticamente com base em alguns critérios, como marcações *html* de início e fim de texto e marcações de parágrafos, para que se possa extrair apenas o texto da notícia. Nesta etapa, informações irrelevantes como *tags html*, publicidade, chamadas para outros artigos e nomes dos autores ou tradutores são removidas do texto coletado.

A terceira etapa consiste em localizar e captar a *url* da versão original da notícia, em português, localizada junto ao texto traduzido para o inglês. Assim, no mesmo instante que o texto em inglês é coletado, também é coletada a *url* da versão original desse texto. A *url* recuperada para a notícia em português é acessada e o texto do artigo é extraído e limpo, assim como realizado para os textos em inglês.

As três etapas iniciais do Módulo coletor lidam com a tarefa de acessar páginas da web. Essa tarefa foi executada utilizando a biblioteca Json.

Por fim, com o par de textos coletado e limpo, cada texto é salvo em um arquivo de texto simples (com a extensão *txt*), com um código numeral padrão. Também é criado um registro no banco de dados para que os pares de textos sejam relacionados, o que é indispensável para as próximas etapas do aprendiz. O banco de dados utilizado nessa etapa foi desenvolvido em PostGre, em linguagem MySQL.

A execução do Coletor termina quando todos os conteúdos de notícias correspondentes às *urls* extraídas na primeira etapa, juntamente com suas versões paralelas, forem coletados, salvos e registrados no banco de dados. Nesse ponto, todo esse conteúdo já está pronto para a fase de pré-processamento do NEPaL.

Um exemplo de um par de textos paralelos gerado como saída do módulo Coletor é apresentado na Tabela 3.1[27]. Nesse exemplo, os textos estão exatamente como nas páginas *web* de onde foram extraídos, sem passar por limpeza ou alteração de formato.

Tabela 3.1 - Exemplo de par de textos paralelos gerado como saída do módulo Coletor. Fonte: www1.folha.uol.com.br.

Texto original em português	Texto traduzido para o inglês
Um imóvel em formato de disco voador que já custou mais de R$ 1 milhão e cuja entrega já está dois anos atrasada pode acabar sem utilidade em Varginha, cidade do sul de Minas que ficou conhecida em todo país há quase 20 anos após a suposta visita de um extraterrestre.	A building shaped like a flying saucer in Varginha, Minas Gerais, which has already cost more than R$ 1 million (US$287,000) and is two years behind schedule, may end up without any official purpose. Varginha became famous nearly twenty years ago after a supposed alien landing.

3.2 Módulo Pré-processador

O Módulo Pré-processador do NEPaL é responsável pelo pré-processamento de todo conteúdo coletado anteriormente pelo módulo Coletor, ou seja, pela geração de dados úteis para o processamento principal (geração de candidatas) realizado no módulo Processador, que será descrito na seção 3.3. Nessa etapa são realizadas várias tarefas que foram agrupadas em três fases para melhor entendimento aqui nesse texto: a) fase de limpeza e conversão; b) fase de alinhamento e etiquetação e c) fase de geração dos dicionários bilíngues.

O módulo Pré-processador recebe como entrada os pares de textos coletados pelo módulo Coletor e gera como saída dois dicionários, um contendo traduções pt-

[27] Texto original em português extraído de: http://www1.folha.uol.com.br/cotidiano/2015/08/1669747-museu-em-formato-de-disco-voador-pode-ficar-sem-utilidade-em-varginha.shtml. Texto traduzido para o inglês extraído de: http://www1.folha.uol.com.br/internacional/en/brazil/2015/08/1670117-museum-shaped-like-flying-saucer-in-minas-gerais-may-end-up-without-purpose.shtml.

en e outro contendo traduções en-pt, seguidas pelo total de ocorrências de cada tradução.

Para tanto, primeiramente, os pares de textos paralelos são recuperados por meio de seu registro no banco de dados e carregados no módulo Pré-processador. Em seguida, os textos passam por um processo de limpeza e tratamento de caracteres especiais. Esse processo é realizado por meio do uso de expressões regulares que realizam: (1) substituições de caracteres especiais por suas ocorrências entre aspas (por exemplo, a vírgula é substituída por sua ocorrência entre aspas *","*), (2) exclusão de linhas em branco e (3) substituição da crase pela letra *a*. Esses processos são necessários para facilitar o alinhamento (no caso da exclusão de linhas em branco) e para garantir que os textos resultantes sejam interpretados corretamente mais adiante no processamento, já que os caracteres vírgula, cifrão, porcentagem, parênteses, colchetes, chaves, aspas e apóstrofo são caracteres reservados no Weka (HALL et al., 2009), ferramenta utilizada na geração do modelo do módulo Promotor, como será explicado na seção 3.4.

Após o pré-processamento, os caracteres restantes são convertidos para minúsculos, pois a existência de palavras iguais no corpus, escritas com variações entre caracteres maiúsculos e minúsculos (como *"PAULISTA"* e *"paulista"*) ou ainda apenas com a inicial maiúscula (como *"Sargento"* e *"sargento"*) afeta o resultado final do processamento do NEPaL. Na sequência, os textos modificados são salvos nos arquivos de texto substituindo os arquivos originais utilizados no início do pré-processamento.

Na segunda fase do pré-processamento, os pares de arquivos gravados na fase anterior são alinhados sentencialmente por meio de uma versão adaptada do alinhador sentencial do PorTAl[28]. Esse alinhamento é necessário devido às diferenças encontradas entre os conteúdos coletados para língua fonte e língua alvo, como conteúdos explicativos que não ocorrem em uma das versões, geralmente na versão em inglês.

Ao final do alinhamento, cada sentença do texto fonte está alinhada com sua sentença correspondente no texto alvo. Sentenças sem uma sentença paralela correspondente são descartadas, o que ocorre principalmente quando o conteúdo de um dos arquivos do par é superior em número de sentenças ao outro. Assim, ao final

[28] Modelos para en-pt disponíveis em Downloads. Disponível em: http://www.lalic.dc.ufscar.br/portal/. Acesso em: 22/01/2016.

desse processo, os arquivos de texto resultantes contêm o mesmo número de sentenças, as quais são paralelas (a primeira sentença no arquivo fonte é paralela à primeira sentença no arquivo alvo e assim por diante). Esses arquivos são armazenados em um diretório auxiliar onde também são gravadas outras informações referentes à execução da ferramenta de alinhamento sentencial.

Assim que se encerra o alinhamento sentencial de todos os arquivos do corpus, seus conteúdos são concatenados e armazenados em um único par de arquivos paralelos, ou seja, os conteúdos de todos os arquivos fonte são copiados para um determinado arquivo (f) e os conteúdos de todos os arquivos alvo para outro arquivo (a) gerando, assim, um corpus paralelo bilíngue alinhado sentencialmente (f-a). Também no final dessa fase são geradas as etiquetas de *part-of-speech* (PoS), as quais são armazenadas em arquivos e posteriormente utilizadas como *features* no arquivo ARFF utilizado na etapa de processamento (esse processo será descrito na seção 3.3). As etiquetas de PoS são geradas automaticamente através do etiquetador de PoS do PorTAl (VIEIRA e CASELI, 2011), o qual processa todo o corpus gerando as etiquetas de PoS para todas as palavras.

Na terceira fase do pré-processamento dos dados, os arquivos com o corpus paralelo bilíngue alinhado sentencialmente e etiquetado são submetidos a uma série de *scripts* do pacote Moses (KOEHN et al., 2007) com o objetivo de criar, entre outros, quatro arquivos fundamentais para a geração dos dicionários bilíngues.

Primeiramente, o corpus paralelo é *tokenizado* (com a inserção de espaços entre palavras e caracteres de pontuação) via script tokenizer.perl e as sentenças longas (acima de 80 palavras) são removidas do corpus pelo script clean-corpus-n.perl. Em seguida, o script train-model.perl é executado com a chamada do MGIZA, uma versão *multicore* do GIZA++ (OCH e NEY, 2003), para se obter alinhamentos lexicais. Ao final desse processo são produzidos, além de vários outros arquivos não utilizados neste projeto, quatro arquivos que são úteis na geração dos dicionários bilíngues: (1) *en.vcb* e (2) *pt.vcb*, que representam o vocabulário do corpus indicando a quantidade de *tokens* encontrados e seu número de ocorrências para o inglês (idioma alvo neste caso) e o português (idioma fonte neste caso), respectivamente, e (3) *lex.e2f* e (4) *lex.f2e*, arquivos com os léxicos fonte-alvo e alvo-fonte, respectivamente, com pares de tradução acompanhados da probabilidade de ocorrência do par no corpus.

A Tabela 3.2 traz trechos dos arquivos de vocabulário e léxico gerados pelos scripts citados.

Tabela 3.2 - Trechos dos arquivos de vocabulário e léxico gerados pelos scripts do Moses.

en.vcb			**pt.vcb**			**lex.e2f**		
		Ocorrências			Ocorrências	pt	en	Frequência
1	the	1299	1	a	1551	duráveis	durable	0.25000
2	a	1214	2	o	1528	avanços	advances	0.33333
3	of	1186	3	de	1497	trimestres	quarters	0.25000

Por fim, a geração dos dicionários é feita pelo script gera_dicionario.pl (CASELI, 2007) cuja função é converter os arquivos de vocabulário (.*vcb*) e léxico (.*e2f* e .*f2e*) em um dicionário bilíngue para que os dados possam auxiliar no alinhamento lexical de novos arquivos.

A Tabela 3.3 traz trechos dos arquivos contendo os dicionários bilíngues pt-en e en-pt gerados pelo script gera_dicionario.pl em que cada par de palavras fonte e alvo é acompanhado pelo número de ocorrências do par no corpus. Nota-se que o número de ocorrências não é o mesmo para os alinhamentos alvo/fonte e fonte/alvo. No caso da terceira linha da Tabela 3.3, por exemplo, o par alinhado *a-the* tem 4256 ocorrências enquanto o par *the-a* tem 4215 ocorrências. Isso ocorre devido às ambiguidades apresentadas no idioma alvo ou às variações lexicais inerentes ao processo de tradução.

Tabela 3.3 - Trechos dos dicionários bilíngue pt-en e en-pt gerados pelo script gera_dicionario.pl.

dicionário bilíngue pt-en			**dicionário bilíngue en-pt**		
		Ocorrências			Ocorrências
a	the	4256	the	a	4215
o	the	3862	the	o	3839
e	and	2984	of	the	2934
de	of	2840	and	e	2894

3.3 Módulo Processador

O módulo Processador do NEPaL é responsável por produzir, a partir dos dicionários gerados pelo módulo Pré-processador, um arquivo com pares de candidatos a paráfrases (ou seja, instâncias candidatas à crenças[29]) em português, ordenados pela probabilidade de serem paráfrases. Esse módulo também gera um arquivo ARFF (como descrito na seção 3.3.1), para ser avaliado pelo módulo Promotor no qual os melhores candidatos são promovidos à crença. Primeiramente, os dicionários passam pelos processos de eliminação de *stopwords* e lematização. As *stopwords* são palavras de grande ocorrência em um corpus, como os *tokens o, a, do, de*. O processo de eliminação de *stopwords* consiste em eliminar palavras ou grupos de palavras que pertençam a classes fechadas, ou seja, classes formadas palavras com pouca variação, como conjunções, artigos e preposições, classes estas que são irrelevantes neste trabalho, além de que, tais exclusões, facilitam o processamento no decorrer do sistema. Esse processo é realizado para ambos os idiomas, português e inglês. Para tanto, foram utilizadas duas *stoplists*: uma com *stopwords* em português e outra em inglês.

Após a remoção das *stopwords*, as palavras restantes foram lematizadas. Esse processo consiste em reduzir as palavras ao seu lema, isto é, os verbos são reduzidos para sua forma no infinitivo impessoal, enquanto as demais palavras flexionadas (substantivos e adjetivos) são reduzidas para suas formas no masculino singular. A lematização foi realizada com o objetivo de agrupar todas as flexões de uma palavra em um único *token*. Assim, é possível evitar que a frequência de certas palavras seja baixa por conta das inúmeras variações que um lema pode sofrer. Para a lematização foi utilizado o Lematizador V2[30]. Vale mencionar que optou-se por realizar a lematização das entradas do dicionário e não do corpus paralelo antes do alinhamento lexical porque, como demonstrado em experimentos prévios Caseli (2007), o alinhamento lexical de textos paralelos envolvendo os idiomas português e inglês apresentou melhores resultados quando foram consideradas formas superficiais do que quando foram considerados os lemas das palavras. Depois da

[29] Crenças, nesse caso, são instâncias promovidas por um sistema de AM.
[30] Disponível em: http://www.icmc.usp.br/pessoas/taspardo/sucinto/resources.html. Acesso em 22/05/2014.

lematização, as entradas com ocorrências iguais foram somadas, por exemplo, imagine como entradas os casos (a) *casa/house/10* (palavra em português/palavra em inglês/nº de ocorrências) e (b) *casas/houses/15*. Após a lematização, (b) se torna *casa/house/15*, mas como já existe um caso com as mesmas entradas (*casa/house*) as frequências são somadas mantendo no dicionário apenas uma entrada: *casa/house/25*.

A Tabela 3.4 traz trechos dos arquivos contendo os dicionários bilíngue pt-en e en-pt após a remoção de *stopwords* e a lematização.

Tabela 3.4 - Trechos dos dicionários bilíngue pt-en e en-pt após a remoção de *stopwords* e a lematização.

dicionário bilíngue pt-en			**dicionário bilíngue en-pt**		
		Ocorrências			Ocorrências
mercado	market	87	region	região	95
polícia	police	67	quarter	trimestre	87
dinheiro	money	65	economy	economia	84

Após a exclusão de *stopwords* e a lematização, os itens restantes são submetidos ao processamento propriamente dito. O Processador do NEPaL foi construído para produzir pares de paráfrases em português utilizando um idioma como pivô, seguindo o método de Bannard e Callison-Burch (2005). A essência desse método é alinhar palavras de um corpus paralelo bilíngue e equacionar diferentes palavras no idioma alvo que estão alinhadas com a mesma palavra, no idioma pivô. No caso deste trabalho, o corpus paralelo bilíngue foi substituído pelos dicionários bilíngues.

Para cada alinhamento resultante do processamento, são geradas quádruplas formadas por *palavra_fonte1*, *palavra_alvo*, *palavra_fonte2* e *probabilidade_par*. Neste caso, em comparação com o método de Bannard e Callison-Burch (2005), pode-se afirmar que *palavra_fonte1* (em pt) foi alinhada com *palavra_alvo* (em en), que por sua vez foi alinhada novamente com *palavra_fonte2* (também em pt e diferente de *palavra_fonte1*). O cálculo probabilístico efetuado entre *palavra_fonte1* e *palavra_fonte2* gera o resultado e o expressa em *probabilidade_par*. A Figura 3.2 traz alguns exemplos dessas quádruplas. Cada um desses alinhamentos entre *palavra_fonte1* e *palavra_fonte2* são considerados candidatos a paráfrase.

```
missão,mission,ajudar,0.820
moradia,house,abrigar,1.639
término,subway,pronto,11.111
declarar,said,dizer,44.060
mandatário,rousseff,dilma,37.298
responsabilidade,supervision,coordenação,1.304
```

Figura 3.2 - Candidatos a paráfrases em português no formato: palavra_fonte1,palavra_alvo,palavra_fonte2,probabilidade_par.

Todas as quádruplas geradas nesta etapa do processamento são novamente processadas e farão parte do arquivo utilizado pelo módulo Promotor para promover as instâncias à crença. Para permitir a promoção automática dessas instâncias a crenças no módulo Promotor, utilizou-se Aprendizado de Máquina (AM) por meio da ferramenta Weka. Para tanto, o arquivo utilizado pelo módulo Promotor deve estar no formato ARFF. Na geração do arquivo ARFF (apresentado na subseção 3.3.1), se tornam os atributos p_fonte1, p_alvo, p_fonte2 e probabilidade.

3.3.1 Geração do arquivo ARFF

O padrão ARFF (*Attribute-Relation File Format*) foi desenvolvido pelo projeto de AM do Departamento de Ciência da Computação da Universidade de Waikato, localizada em Hamilton, na Nova Zelândia. É um arquivo texto que descreve uma lista de instâncias que compartilham o mesmo conjunto de atributos e é usado como entrada para o Weka (HALL et al., 2009), um *toolkit* de mineração de dados utilizado neste trabalho.

Neste projeto, foram definidas 19 *features* (atributos) para o aprendizado das paráfrases. Essas *features* foram elaboradas para serem usadas no AM supervisionado, ou seja, ser utilizadas para classificar exemplos positivos e negativos e, consequentemente, permitir que o aprendiz (nesse caso, o Módulo Promotor) se torne capaz de classificar instâncias corretamente. As 19 *features* foram elaboradas conforme descritos a seguir:

1. p_fonte1: atributo de texto, representando a palavra_fonte1 que ocorre em uma instância (par de paráfrases).
2. p_alvo: atributo de texto, representando a palavra_alvo que ocorre em uma instância.
3. p_fonte2: atributo de texto, representando a palavra_fonte2 que ocorre em uma instância.

4. antes3_fonte: atributo de texto, composto pelo *token* que ocorre 3 posições antes da p_fonte1 ou p_fonte2.
5. antes2_fonte: atributo de texto, composto pelo *token* que ocorre 2 posições antes da p_fonte1 ou p_fonte2.
6. antes1_fonte: atributo de texto, composto pelo *token* que ocorre imediatamente antes da p_fonte1 ou p_fonte2.
7. depois1_fonte: atributo de texto, composto pelo *token* que ocorre imediatamente após a p_fonte1 ou p_fonte2.
8. depois2_fonte: atributo de texto, composto pelo *token* que ocorre 2 posições após a p_fonte1 ou p_fonte2.
9. depois3_fonte: atributo de texto, composto pelo *token* que ocorre 3 posições após a p_fonte1 ou p_fonte2.
10. postag_p_fonte1: atributo de texto formado pela etiqueta de PoS, como v (verbo), adj (adjetivo), n (substantivo) e prn (pronome), à qual corresponde a p_fonte1.
11. postag_p_fonte2: atributo de texto formado pela etiqueta de PoS correspondente a p_fonte2.
12. postag_antes3_fonte: atributo de texto, composto pela etiqueta de PoS do *token* que ocorre 3 posições antes de p_fonte1 ou p_fonte2.
13. postag_antes2_fonte: atributo de texto, composto pela etiqueta de PoS do *token* que ocorre 2 posições antes da p_fonte1 ou p_fonte2.
14. postag_antes1_fonte: atributo de texto, composto pela etiqueta de PoS do *token* que ocorre imediatamente antes da p_fonte1 ou p_fonte2.
15. postag_depois1_fonte: atributo de texto, composto pela etiqueta de PoS do *token* que ocorre imediatamente após p_fonte1 ou p_fonte2.
16. postag_depois2_fonte: atributo de texto, composto pela etiqueta de PoS do *token* que ocorre 2 posições após p_fonte1 ou p_fonte2.
17. postag_depois3_fonte: atributo de texto, composto pela etiqueta de PoS do *token* que ocorre 3 posições após p_fonte1 ou p_fonte2.
18. LCSR: valor de *Longest Common Subsequence Ratio* (ou LCSR), medida que mede se duas palavras são cognatas, calculada através da divisão da maior subsequência comum entre duas palavras pelo tamanho da maior palavra (HSU e DE, 1984). É um atributo numérico, variando entre 0 e 1, calculado entre p_fonte1 e p_fonte2.

19. probabilidade: atributo numérico, variando entre 0 e 100. Esse valor provém da probabilidade já calculada anteriormente com base no alinhamento lexical, conforme proposto por (BANNARD e CALLISON-BURCH, 2005).

As features de contexto, ou seja, *antes3_fonte*, *antes2_fonte*, *antes1_fonte*, *depois1_fonte*, *depois2_fonte*, *depois3_fonte*, *postag_p_fonte1*, *postag_p_fonte2*, *postag_antes3_fonte*, *postag_antes2_fonte*, *postag_antes1_fonte*, *postag_depois1_fonte*, *postag_depois2_fonte* e *postag_depois3_fonte* são extraídas diretamente da sentença original onde uma das palavras do par de candidatas ocorre, ou seja, onde ocorre *p_fonte1* ou *p_fonte2*. Elas se referem a p_fonte1 ou p_fonte2, dependendo de qual ocorre na sentença.

Além dos atributos descritos acima, também fazem parte do arquivo ARFF linhas de comentários (iniciadas com %) para cada instância: uma contendo a numeração do candidato, seguida pelo nome do sistema ou do avaliador que classificou a instância e o par de candidatas; e outra contendo uma sentença do corpus original onde uma das palavras do par de candidatas ocorreu precedida das posições de início e fim da ocorrência da candidata na sentença.

A Figura 3.3 apresenta três exemplos de instâncias: 152, 184 e 22764, respectivamente. Em todos os casos, a primeira das três linhas de comentários (com o símbolo de % no início da linha) traz os números identificadores das instâncias seguidos pelo nome do sistema que as gerou e pelo par de candidatas ("*previdência*" e "*sistema*", no primeiro exemplo, "*haddad*" e "*fernando*", no segundo exemplo e "*independente*" e "*autônomo*", no terceiro exemplo). Já na segunda linha de comentário de cada exemplo observa-se a posição de início e fim da ocorrência de uma das candidatas na sentença original do *corpus* que vem logo em seguida. A terceira linha de comentário traz as *features* nominais (que nesse caso não foram utilizadas no treinamento do Promotor). Por fim, a quarta linha de cada exemplo traz as *features* efetivamente usadas: valor de LCSR e probabilidade de serem paráfrases. Apesar de inicialmente terem sido definidas 19 atributos, foram utilizados apenas LCSR e probabilidade porque, em experimentos com três algoritmos (Naïve Bayes, SVM e J48), constatou-se que os resultados se tornam dependentes do corpus, inviabilizando o aprendizado de máquina.

A partir da geração deste arquivo, as instâncias estão prontas para serem avaliadas e, consequentemente, serem ou não promovidas a crenças.

```
@relation parafrase

@attribute LCSR NUMERIC
@attribute probabilidade NUMERIC
@attribute is_parafrase {yes,no}

@data

% cand=152 nepal previdência sistema
% 43 43 primeiramente , cabe lembrar que temos estabilizadores automáticos da demanda que ajudarão a
% previdência,system,sistema,público,",'"',de+o,e,de,inúmero,n,n,adj,cm,pr+det,cnjcoo,pr,adj
0.091,10.959,?

% cand=184 nepal haddad fernando
% 15 15 o governo tião viana diz que os haitianos não querem ficar no acre . haddad cobra planejament
% haddad,fernando,fernando,em+o,acre,.,cobrar,planejamento,de+o,NC,NC,pr+det,n,sent,v,n,pr+det
0.125,11.507,?

% cand=22764 nepal independente autônomo
% 23 23 ponto para o governo , pois , pelo ambiente atual , o cenário era perfeito para a oposição se
% independente,independent,autônomo,juntar,a+o,grupo,e,colocar,mais,adj,adj,v,pr+det,n,cnjcoo,v,adv
0.083,50.000,?
```

Figura 3.3 - Trecho do arquivo ARFF gerado pelo processador.

3.4 Módulo Promotor

O módulo Promotor é o último módulo do NEPaL e é responsável por promover as instâncias geradas pelo módulo Processador à crenças. Neste módulo, por meio do *toolkit* Weka (HALL et al., 2009), algoritmos de aprendizado de máquina podem ser executados para classificar cada instância gerada pelo Processador com base em um modelo previamente treinado.

O Weka possui vários algoritmos de aprendizado de máquina baseados em diferentes métodos, como Árvores de Decisão e modelos probabilísticos. Para a criação do modelo treinado utilizado para classificar as instâncias, três algoritmos disponíveis no Weka foram testados: o SVM, por meio da biblioteca LibSVM (CHANG e LIN, 2011), o Naïve Bayes (LEWIS, 1992) e a árvore de decisão J48, baseada no algoritmo C4.5 (QUINLAN, 1993).

Para tanto, um conjunto de treinamento foi gerado com 1800 instâncias avaliadas manualmente por dois juízes falantes nativos do português com o auxílio da ferramenta NEPaLE (TEIXEIRA et al., 2015). Essa ferramenta foi desenvolvida pelo aluno de Iniciação Científica Rafael Teixeira, membro do LALIC[31], com o intuito de apoiar o processo de anotação de candidatas a paráfrases, sempre considerando o contexto de uma sentença na qual uma das candidatas ocorre. Por meio da

[31] Laboratório de Linguística e Inteligência Computacional localizado no Departamento de Computação – UFSCar.

NEPaLE[32], cada juiz foi instruído a anotar o par de candidatas como “SIM” (é uma paráfrase) se ao substituir uma palavra pela outra na sentença o sentido é preservado dado o contexto, e como “NÃO” (não é uma paráfrase) caso essa substituição acarrete alguma alteração no sentido original da sentença. A tela de anotação da NEPaLE pode ser visualizada na Figura 3.4. A NEPaLE recebe como entrada um arquivo ARFF e gera como saída o mesmo arquivo de entrada, com o nome do anotador (em uma das linhas de comentário) e a classe (yes ou no) atribuída à instância anotada (como o último atributo). Um trecho de um arquivo ARFF após a anotação manual realizada usando a NEPaLE pode ser visualizado na Figura 3.5.

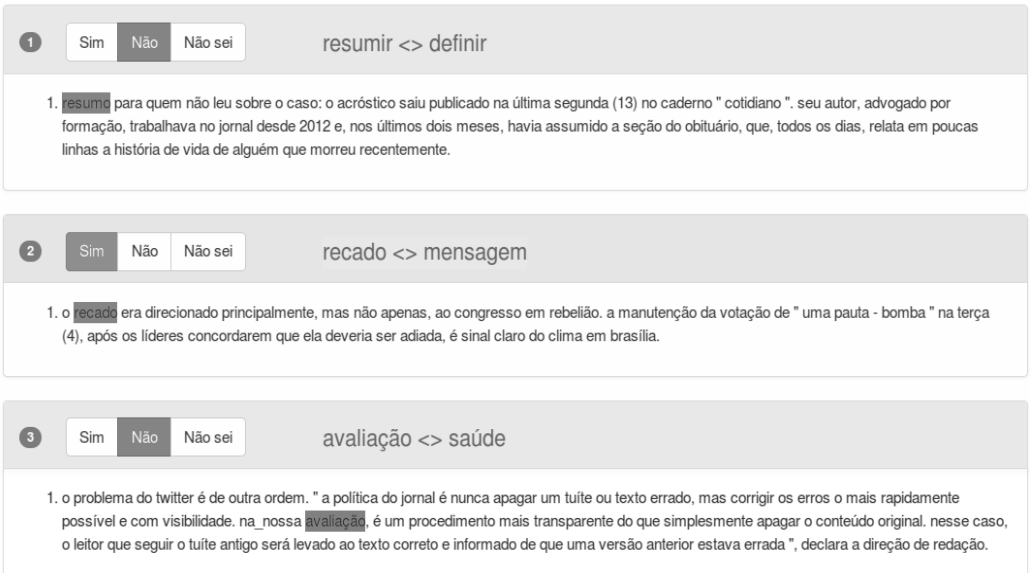


Figura 3.4 - Tela de anotação da NEPaLE usada para gerar o corpus de treinamento par ao Promotor-0.

```
@data

% cand=220987 helena ramo campo
% 26 26 a condição material de vida da população vai
% ramo,field,campo,e,em,outro,de+o,estado,ficar,n,n,
0.400,16.982,no

% cand=268522 helena escritório sala
% 32 32 o detento só consegue falar com sua família
% escritório,office,sala,difícil,chegar,a+o,de+o,cre
0.100,22.159,yes
```

Figura 3.5 – Trecho de arquivo arff anotado.

[32] Disponível em http://www.lalic.dc.ufscar.br/avaliacao-nepal/nepal/login.php. Acesso em: 04/02/2016.

A concordância obtida por esses juízes na tarefa de anotação do corpus de treinamento foi de κ= 0,85 (CARLETTA, 1996), considerada como boa de acordo com os valores relatados na literatura (BARZILAY e McKEOWN, 2001). A partir desses dados de treinamento, um modelo foi gerado para ser usado como o Promotor inicial do NEPaL: o Promotor-0, como descrito a seguir.

3.4.1 Experimentos para geração do Promotor-0

Como mencionado anteriormente, na seção 3.3.1, os arquivos ARFF gerados neste projeto contêm 19 atributos (*features*). Para se tentar determinar qual a melhor configuração para o AMSF, em alguns experimentos foi considerado o número total de atributos, ou seja, todos 19 atributos e, em outros testes foram considerados apenas os atributos numéricos, ou seja, LCSR e probabilidade de serem paráfrases.

Nos experimentos com os 19 atributos (conjunto *C1*), em qualquer um dos algoritmos, pôde-se notar que os modelos aprendidos dependiam muito dos atributos nominais. Dessa forma, o processo se mostrou totalmente dependente do corpus utilizado para gerar os arquivos de treinamento, tornando inviável utilizar todos os atributos para o treinamento do Promotor. Por esse motivo, optou-se por utilizar apenas os atributos numéricos das instâncias para treinar o Promotor-0.

Nos experimentos considerando apenas os atributos numéricos, foram utilizados 3 conjuntos de testes chamados de *C2, C3 e C4. C1* e *C2* possuem conteúdos distintos, já *C3* é uma combinação de *C1* e *C2* e *C4* possui o conteúdo de *C3* além de mais instâncias.

O conjunto C1 é formado por 429 instâncias que foram avaliadas por dois juízes falantes nativos do português. Essas instâncias foram divididas em dois subconjuntos de 255 instâncias, onde 81 instâncias são iguais nos dois subconjuntos e 174 são distintas. Das 255 instâncias do primeiro subconjunto, o juiz1 avaliou 20 como paráfrases enquanto das 255 instâncias do segundo subconjunto, o juiz2 considerou 55 como paráfrases. As 81 instâncias compartilhadas entre os dois juízes só foram consideradas paráfrases quando ambos avaliaram como paráfrase, totalizando 10 instâncias. Como resultado tem-se que C1 contém 65 instâncias avaliadas como paráfrases (classe YES) e 364 classificadas como não paráfrases (classe NO).

Já o conjunto C2 é composto por 486 instâncias, as quais foram divididas em dois subconjuntos contendo 285 instâncias, sendo 84 instâncias iguais nos dois conjuntos e 201 instâncias distintas. Assim como no conjunto C1, cada juiz avaliou um subconjunto de instâncias do C2. O juiz1 avaliou 72 instâncias como paráfrase, já o juiz2, 107. Considerando as instâncias iguais avaliadas como paráfrases por ambos os juízes, tem-se um total de 137 instâncias consideradas paráfrase (classe YES) e 349 não paráfrases (classe NO).

Ambos os conjuntos C1 e C2 foram criados visando testar o desbalanceamento das classes positiva e negativa, por isso o número de instâncias positivas e negativas de cada um são distintos.

Para lidar com o desbalanceamento entre as classes positiva (YES) e negativa (NO), a partir de C1 e C2 outros conjuntos foram gerados. Primeiro, gerou-se conjuntos sem a replicação de instâncias: C1_130 (contendo 65 instâncias positivas e 65 instâncias negativas selecionadas aleatoriamente) e C2_274 (contendo 137 instâncias positivas e 137 instâncias negativas selecionadas aleatoriamente). Outra estratégia utilizada para lidar com o desbalanceamento das classes foi replicar as instâncias positivas o que resultou nos conjuntos: C1_412 (contendo 206 instâncias positivas e 206 instâncias negativas) e C2_900 (contendo 450 instâncias positivas e 450 instâncias negativas).

Os conjuntos de teste C3_404 (contendo 202 instâncias positivas e 202 instâncias negativas) e C3_1306 (contendo 653 instâncias positivas e 653 instâncias negativas) foram formados com instâncias selecionadas aleatoriamente dos quatro conjuntos de testes mencionados anteriormente (C1_130, C1_412, C2_274 e C2_900).

O conjunto de testes C4_1800 (contendo 900 instâncias positivas e 900 instâncias negativas) foi formado utilizando todas as instâncias contidas no conjunto C4, com algumas instâncias positivas duplicadas. Isso foi feito com o objetivo de igualar o número de instâncias positivas e negativas no conjunto de testes.

A Tabela 3.5 resume as quantidades de instâncias nos conjuntos de treinamento.

A partir desses conjuntos, foram realizados 7 experimentos, todos feitos utilizando a opção de *cross-validation (10-folds)*. Cada experimentos foi feito como segue.

Tabela 3.5 - Quantidade de instâncias nas classes positiva (YES) e negativa (NO) nos conjuntos de treinamento.

Conjuntos	YES	NO
C1	65	364
C2	137	349
C1_130	65	65
C2_274	137	137
C1_412	206	206
C2_900	450	450
C3_404	202	202
C3_1306	653	653
C4_1800	900	900

1. C1 com 130 instâncias (C1_130) – Com esse conjunto de dados foram obtidos os seguintes resultados: 81,53% das instâncias foram classificadas corretamente e 18,47% incorretamente com o classificador Naïve Bayes. Já com SVM e J48 os resultados foram exatamente os mesmos, 88,46% classificadas corretamente e 11,54% incorretamente.
2. C1 com 412 instâncias (C1_412) – Neste teste foram obtidas 71,60% de instâncias classificadas corretamente e 28,40% incorretamente, com Naïve Bayes, 82,04% classificadas corretamente e 17,96% incorretamente com o SVM e 93,69% classificadas corretamente e 6,31% classificadas incorretamente com o J48. Neste teste é possível notar que J48 obteve resultados excelentes.
3. C2 com 274 instâncias (C2_274) – Com estes dados os resultados obtidos são: 59,85% das instâncias classificadas corretamente e 40,15% incorretamente, com Naïve Bayes, 68,25% classificadas corretamente e 31,75% incorretamente, com o SVM e 70,06% classificadas corretamente e 29,94% classificadas incorretamente, com o J48.
4. C2 com 900 instâncias (C2_900) – Com 900 instâncias os resultados para o conjunto C2 são melhores que com 274 instâncias: 69,11% classificadas corretamente e 30,89% incorretamente, com Naïve Bayes, 75,55% classificadas corretamente e 24,45% incorretamente, com o SVM e 79,67% classificadas corretamente e 20,33% classificadas incorretamente, com o J48.

5. C3 com 404 instâncias (C3_404) – Nesse caso, 71,29% das instâncias foram classificadas corretamente e 28,71% incorretamente com Naïve Bayes, 69,80% classificadas corretamente e 30,20% incorretamente com o SVM e 70,30% classificadas corretamente e 29,70% classificadas incorretamente, com o J48.
6. C3 com 1306 instâncias (C3_1306) – Assim como no conjunto C2, o conjunto C3 atinge, na média entre os classificadores, melhores resultados quando mais instâncias são utilizadas: 70,29% classificadas corretamente e 29,71% incorretamente com Naïve Bayes, 74,20% classificadas corretamente e 25,80% incorretamente com o SVM e 84,69% classificadas corretamente e 15,31% classificadas incorretamente, com o J48.
7. C4 com 1800 instâncias (C4_1800) – Entre todos os conjuntos, acredita-se que os resultados desse conjunto sejam os mais confiáveis pelo fato de possuir mais instâncias em relação aos outros conjuntos. Nesse teste 59,15% classificadas corretamente e 40,86% incorretamente com Naïve Bayes, 64,35% classificadas corretamente e 35,65% incorretamente com o SVM e 89,74% classificadas corretamente e 10,26% classificadas incorretamente, com o J48.

A tabela 3.6 traz um resumo dos resultados obtidos por cada conjunto de treinamento testado, onde ICC representa a porcentagem de instâncias classificadas corretamente e ICI representa a porcentagem de instâncias classificadas incorretamente.

Tabela 3.6 - Resumo dos testes com algoritmos para criação do Promotor-0.

Conjuntos	Naïve Bayes		SVM		J48	
	ICC	ICI	ICC	ICI	ICC	ICI
C1_130	**81,53%**	18,47%	**88,46%**	11,54%	**88,46%**	11,54%
C1_412	71,60%	28,40%	**82,04%**	17,96%	**93,69%**	6,31%
C2_274	59,85%	40,15%	68,25%	31,75%	70,06%	29,94%
C2_900	69,11%	30,89%	75,55%	24,45%	79,67%	20,33%
C3_404	71,29%	28,71%	69,80%	30,20%	70,30%	29,70%
C3_1306	70,29%	29,71%	74,20%	25,80%	**84,69%**	15,31%
C4_1800	59,15%	40,86%	64,35%	35,65%	**89,74%**	10,26%

A Seguir, o gráfico expresso na Figura 3.6 mostra os valores obtidos nos experimentos com os classificadores.

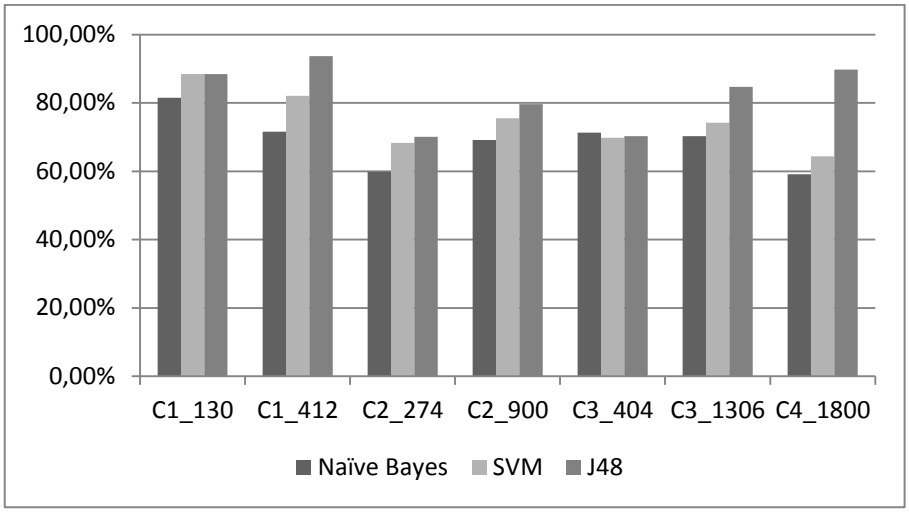


Figura 3.6 – Gráfico de desempenho com algoritmos para criação do Promotor-0.

Após a realização desses experimentos, optou-se por utilizar como conjunto de dados para treinamento do Promotor-0 o C4_1800 e o J48 como algoritmo. Apesar de um conjunto de testes obter resultados melhores (como C1_412 com J48) e outros conjuntos atingir resultados muito próximos (como C1_130 com SVM e J48), esse conjunto foi escolhido, pois além de atingir uma boa porcentagem de instâncias classificadas corretamente, também possui um número maior de instâncias em relação aos outros testes. O maior número de instâncias no corpus de treinamento é uma característica desejável para se evitar o *overfitting*[33]. Vale mencionar que esperava-se melhores resultados nos experimentos com SVM, mas em quase todos os experimentos o algoritmo J48 se mostrou superior. Talvez esses resultados reflitam a melhor capacidade do J48 em lidar com um conjunto com poucas instâncias.

Com o Promotor-0 arquivos ARFF gerados pelo Processador podem ter suas instâncias classificadas como crenças. Por exemplo, um trecho de um arquivo ARFF a ser classificado pelo Promotor pode ser visto na Figura 3.7.

Desse modo, o arquivo com o conjunto de treinamento C4_1800 foi utilizado para gerar o modelo treinado inicial do promotor, o Promotor-0. Com a geração do Promotor-0 o NEPaL está pronto.

Além do Promotor-0, outros dois modelos de Promotor foram criados neste projeto para poder avaliar o aprendizado sem-fim: o Promotor-1 e o Promotor-2. Mais detalhes desses modelos são apresentados na seção 4.1

[33] *Overfitting* é o termo utilizado no AM quando ocorre um ajuste excessivo do modelo estatístico em relação ao conjunto de dados de treinamento.

```
@relation parafrase

@attribute LCSR NUMERIC
@attribute probabilidade NUMERIC
@attribute is_parafrase {yes,no}

@data


0.083,15.301,?
0.077,10.526,?
0.167,2.000,?
0.222,2.000,?
0.091,0.853,?
0.143,5.732,?
0.286,5.732,?
0.250,1.274,?
```

Figura 3.7 - Trecho do arquivo ARFF a ser classificado pelo Promotor.

Após a geração do arquivo ARFF e o treinamento do Promotor-0 estar concluído, está tudo pronto para o início dos experimentos.

Capítulo 4

EXPERIMENTOS E RESULTADOS

Este capítulo tem como objetivo contextualizar o leitor sobre os experimentos realizados neste trabalho, bem como os resultados obtidos através destes experimentos e as avaliações destes resultados.

Esse capítulo está organizado da seguinte forma. Na seção 4.1 estão descritos os experimentos realizados durante o projeto e os resultados obtidos, enquanto a seção 4.2 traz a análise quantitativa dos resultados obtidos para a avaliação do aprendizado sem-fim e a seção 4.3 uma análise qualitativa dos dados.

4.1 Experimentos

Após o término da construção do NEPaL e a geração do modelo treinado do promotor inicial, o Promotor-0, conforme descrito no Capítulo 3, deu-se início aos experimentos com o NEPaL com o objetivo de verificar se é possível utilizar a estratégia de aprendizado de máquina sem-fim e a internet para aprender paráfrases de forma incremental e automática.

No início dos experimentos, na fase de coleta, as *urls* foram coletadas em períodos alternados da semana, a cada 48 horas, com o objetivo de evitar a coleta de *urls* repetidas e, consequentemente, textos repetidos. Essa pausa evita que o módulo Coletor fique procurando por *urls* novas em meio a muitas já coletadas.

Foi definido que cada iteração do NEPaL teria sempre 40 pares de textos, com o intuito de manter um controle sobre a quantidade de dados processados a cada iteração. Por isso, quando o Coletor atinge 40 *urls* coletadas, o conteúdo textual das 40 *urls* é recuperado, assim como sua tradução (em português, como

descrito no Capítulo 3) e então o módulo Coletor entra em espera. Em seguida, o conteúdo coletado passa a ser processado pelas fases de pré-processamento e as fases seguintes.

Como uma estratégia adotada para tentar melhorar a qualidade do alinhamento lexical, o corpus formado a cada iteração contém o corpus das iterações anteriores, ou seja, a cada iteração, o corpus formado pela coleta de 40 pares de notícias é acrescido ao corpus formado anteriormente, além das 10000 palavras que mais ocorreram nos léxicos produzidos por Caseli (2003). Essa estratégia de concatenação de novos textos coletados ao corpus atual foi adotada uma vez que o tamanho do corpus influencia na qualidade do alinhamento lexical gerado pelo alinhador estatístico GIZA++.

Para o experimento descrito a seguir foram executadas 15 iterações, totalizando 600 pares de notícias coletados. Durante as 15 iterações, o Promotor foi treinado 3 vezes, a cada 5 iterações (200 pares de notícias processados): no início (antes da primeira iteração) gerando o Promotor-0, entre a quinta e a sexta iterações, gerando o Promotor-1, e entre a décima e a décima primeira iterações, gerando o Promotor-2. Todos os treinamentos foram realizados usando o algoritmo J48 do Weka, uma vez que esse foi o algoritmo que apresentou melhores resultados nos experimentos realizados para a geração do Promotor-0, conforme descrito na seção 3.4.1.

Desse modo, as primeiras 5 iterações (iterações de 1 a 5) do NEPaL foram feitas utilizando o Promotor-0 (veja seção 3.4.1) como modelo treinado. Durante essas 5 iterações, foram obtidas 398 candidatas à paráfrase, das quais, 184 foram promovidas à crença pelo Promotor-0 (cerca de 46,23%). Todas as crenças promovidas foram avaliadas por um juiz falante nativo do português, o juiz1, usando a ferramenta NEPaLE.

O juiz1 foi instruído a avaliar todas as crenças produzidas pelo Promotor-0. Das 184 crenças produzidas, 74,46% das crenças promovidas pelo Promotor-0 estavam corretas e 25,54% incorretas. Estes dados estão disponíveis na Tabela 4.1.

Tabela 4.1 - Avaliação das crenças produzidas pelo Promotor-0.

	Total de crenças	**Crenças corretas**	**Crenças incorretas**
Juiz1	184	137 (74,46%)	47 (25,54%)

Após as 5 primeiras iterações, o Promotor foi novamente treinado a partir de instâncias produzidas até então pelo módulo Processador do NEPaL (não anotadas por humanos) e também instâncias utilizadas no treinamento do Promotor-0. Para o treinamento de uma nova versão do Promotor, o Promotor-1, foram utilizadas 2000 instâncias. Destas, 1000 foram anotadas como paráfrases pelos juízes ou promovidas a crença pelo Promotor-0 e 1000 foram anotadas pelos juízes como não paráfrase ou classificadas como não paráfrase pelo Promotor-0.

Após a produção do Promotor-1, mais 5 iterações (iterações de 6 a 10) foram executadas, produzindo um total de 349 instâncias, das quais, 164 foram promovidas à crença (cerca de 46,99%). Vale mencionar que nenhuma instância produzida pelo Promotor-1 foi manipulada por humanos.

Assim como no julgamento das crenças produzidas pelo Promotor-0, todas as crenças foram avaliadas pelo juiz1, que julgou que 127 das crenças promovidas pelo Promotor-1 estão corretas (aproximadamente 77,44%) e 37 (aproximadamente 22,56%) estão incorretas. A Tabela 4.2 apresenta esses resultados.

Tabela 4.2 - Avaliação das crenças produzidas pelo Promotor-1.

	Total de crenças	**Crenças corretas**	**Crenças incorretas**
Juiz1	164	127 (77,44%)	37 (22,56%)

Assim como ao final da quinta iteração, ao final da décima, o Promotor foi novamente treinado, utilizando as crenças e instâncias avaliadas como não paráfrase, instâncias anotadas anteriormente pelos juízes e instâncias classificadas pelo módulo Promotor até a iteração atual. Dessa vez foram utilizadas 2300 instâncias, com a quantidade balanceada de crenças e instâncias classificadas como não paráfrase: 1150 cada.

Essa versão do Promotor é a versão responsável por avaliar as instâncias produzidas pelas 5 últimas iterações do NEPaL (iterações de 11 a 15) e é chamada de Promotor-2.

Durante as 5 iterações utilizando o Promotor-2, foram produzidas um total de 322 instâncias. Destas, 154 (aproximadamente 47,82%) foram promovidas à crença pelo Promotor-2. Assim como no Promotor-1, nenhuma instância produzida pelo Promotor-2 foi manipulada por humanos.

Novamente, o juiz1 foi instruído a avaliar todas as instâncias promovidas pelo Promotor. Nestas avaliações, segundo o juiz1, 133 (aproximadamente 86,36%) das

crenças promovidas pelo Promotor-1 estão corretas e 21 (aproximadamente 13,64%) estão incorretas. A Tabela 4.3 mostra esses números.

No total, durante as 15 iterações, foram geradas 1069 instâncias candidatas à paráfrase, das quais, 502 (cerca de 46,95%) foram promovidas à crença pelas versões dos promotores Promotor-0, Promotor-1 e Promotor-2.

Tabela 4.3 - Avaliação das crenças produzidas pelo Promotor-2.

	Total de crenças	Crenças corretas	Crenças incorretas
Juiz1	154	133 (86,36%)	21 (13,64%)

A Tabela 4.4 apresenta um comparativo entre os resultados das avaliações para cada 5 iterações, ou seja, para cada uma das três versões do Promotor utilizadas nos experimentos. Na tabela 4.4, CAC representa as crenças produzidas avaliadas como corretas pelo juiz enquanto CAI representa as crenças avaliadas como incorretas.

Tabela 4.4 - Avaliação das crenças produzidas pelos Promotores.

	Juiz1			
	CAC	**CAI**	**CAC(%)**	**CAI(%)**
Promotor-0	137	47	74,46	25,54
Promotor-1	127	36	77,44	22,56
Promotor-2	133	21	86,36	13,64
Média	**132,33**	**34,66**	**79,42**	**20,58**

As Figuras 4.1, 4.2 e 4.3 trazem exemplos de paráfrases (crenças) geradas pelos promotores 0, 1 e 2, respectivamente, acompanhadas das sentenças nas quais uma das palavras de cada candidata ocorre e a avaliação atribuída pelo juiz humano.

Figura 4.1 - Crenças geradas pelo Promotor-0.

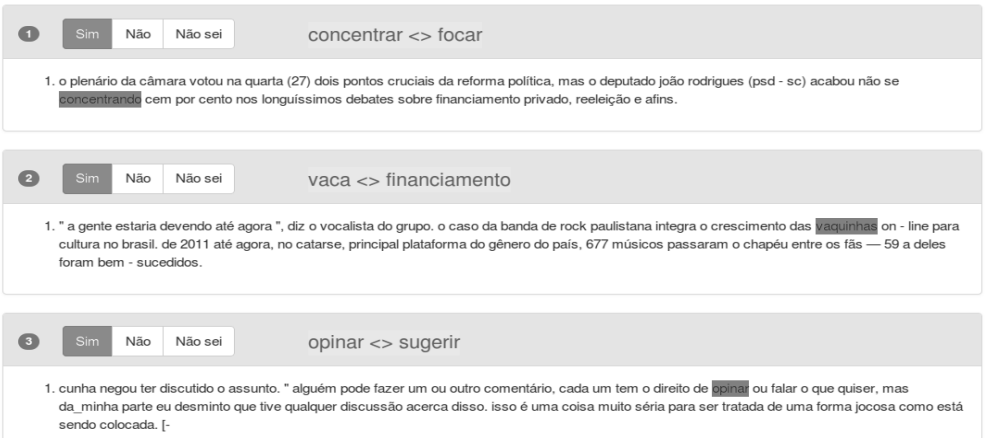


Figura 4.2 - Crenças geradas pelo Promotor-1.

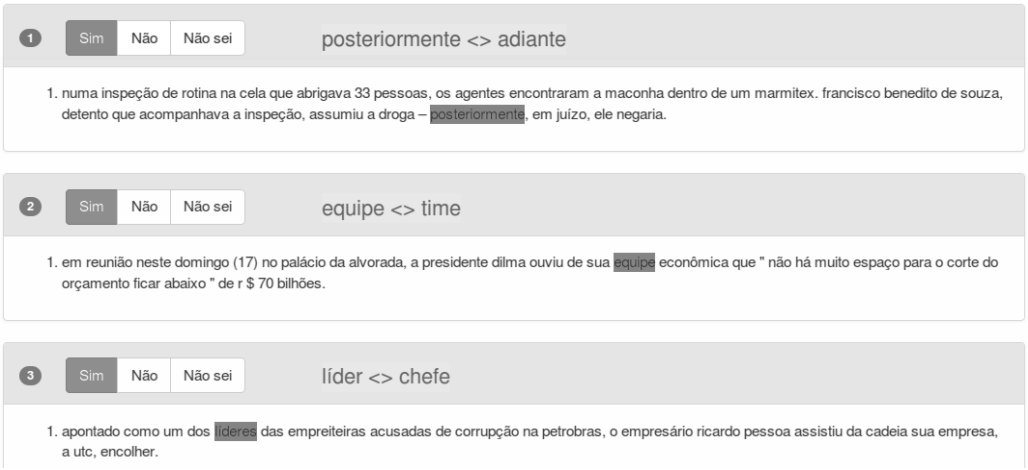
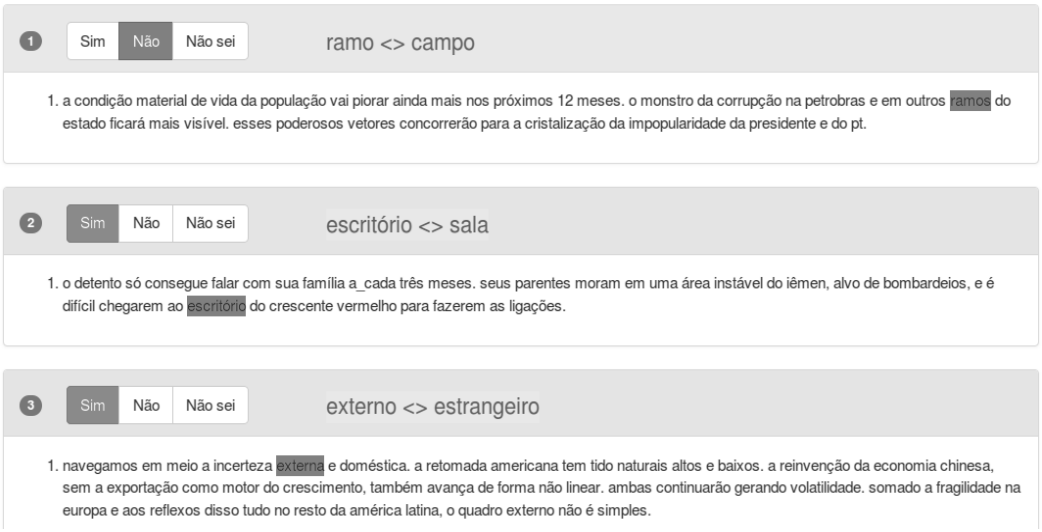


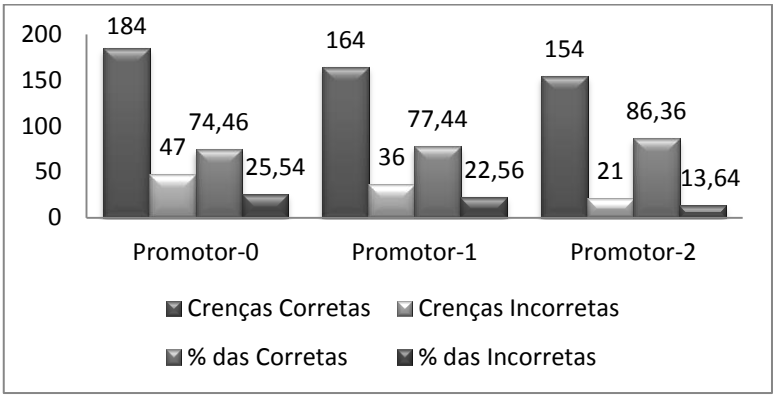
Figura 4.3 - Crenças geradas pelo Promotor-2.



A seguir, na Figura 4.4, o gráfico representa a quantidade de instâncias produzidas e a quantidade de crenças avaliadas como corretas e incorretas a partir do número de instâncias, considerando cada versão do Promotor.

Com a ajuda do gráfico expresso na figura 4.4 é possível notar que durante as iterações, tanto o número de crenças corretas quanto incorretas cai, ou seja, o número de crenças produzidas durante as iterações diminui. É possível notar também que, apesar do número de crenças diminuir, a porcentagem de crenças corretas sempre sobe.

Figura 4.4 - Gráfico da evolução do Promotor.



4.2 Avaliação do Aprendizado de Máquina Sem-Fim

A avaliação da estratégia de AMSF neste projeto foi feita com base nas crenças promovidas durante as 15 iterações executadas na fase de experimentos (veja seção 4.1). A cada 5 iterações, as candidatas promovidas foram avaliadas por um juiz, o juiz1, que avaliou se cada candidata promovida era realmente uma paráfrase ou não. Dessa forma, é possível avaliar se houve melhora no aprendizado através da porcentagem de acerto que cada versão do Promotor obteve em relação às candidatas promovidas.

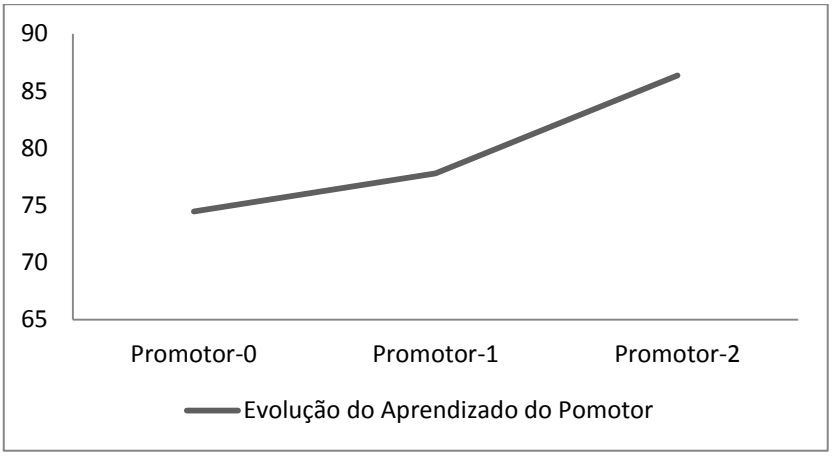
Nas primeiras 5 iterações (iterações de 1 a 5), ou seja, utilizando a versão do promotor Promotor-0, foram promovidas 184 instâncias a crença, das quais 137 (74,46%) foram avaliadas como corretas pelo juiz1.

Nas iterações de 6 a 10 foi utilizada a versão do promotor Promotor-1, gerado a partir das crenças obtidas nas iterações de 1 a 5 (vide seção 4.1). Nessas iterações (de 6 a 10) foram promovidas 164 instâncias a crença, das quais 127 (77,44%) foram avaliadas como corretas pelo juiz1.

Já nas iterações de 11 a 15 foi utilizada a versão do promotor Promotor-2, criado a partir das crenças obtidas nas iterações de 11 a 15, da mesma forma que a versão Promotor-1. Nas iterações de 11 a 15 foram promovidas 154 instâncias a crença e 133 (86,36%) foram avaliadas como corretas pelo juiz1.

A Figura 4.5 exibe um gráfico que mostra a evolução das versões do Promotor: Promotor-1, Promotor-2 e Promotor-3.

Figura 4.5 - Gráfico de Evolução do Promotor: porcentagem de crenças corretas geradas por cada versão do Promotor.



Como pode ser visto na Figura 4.5, existe uma evolução crescente entre as crenças promovidas corretamente desde o Promotor-0 até o Promotor-2, ou seja, a precisão/desempenho das versões do Promotor melhora durante as iterações. Dessa forma é possível afirmar que a aplicação da técnica do AMSF é viável para o aprendizado de paráfrases. Com o tempo, o sistema apresentou uma evolução e melhorou sua capacidade de identificar paráfrases.

Além disso, vale ressaltar que o sistema se manteve constante em relação à porcentagem das instâncias geradas que foram promovidas à crença: 398 instâncias foram geradas nas iterações 1-5 das quais 184 foram promovidas à crença pelo Promotor-0 (cerca de 46,23%); 349 instâncias foram geradas nas iterações 6-10 das quais 164 foram promovidas à crença pelo Promotor-1 (cerca de 46,99%); e 322 instâncias foram geradas nas iterações 11-15 das quais 154 foram promovidas à crença pelo Promotor-2 (aproximadamente 47,83%). Dessa forma é possível notar que o ganho de precisão (visível no gráfico da Figura 4.11) não acarretou perda na cobertura, uma vez que a porcentagem de crenças promovidas em relação ao número de instâncias geradas se manteve próximo a 47% nas três gerações avaliadas.

4.3 Análise qualitativa dos dados

Além dos resultados considerados corretos e incorretos pelo juiz1, existem casos que, em decorrência da limitação da estratégia adotada, o par de candidatas não foi considerado paráfrase. Segundo os juízes, esses casos seriam considerados paráfrases se as candidatas estivessem em outro contexto. Esses casos ocorrem, principalmente, devido à ambiguidade de palavras no idioma pivô e também quando uma das palavras do par faz parte de expressões multipalavras, no idioma alvo.

A seguir alguns desses casos são discutidos para que se tenha uma ideia das limitações e problemas atuais da estratégia adotada no NEPaL. Contudo, uma avaliação quantitativa ou uma análise detalhada desses casos não serão apresentadas neste documento.

Uma das ocorrências onde a ambiguidade do idioma pivô interfere na avaliação dos juízes é apresentada no par *trabalhar*<>f*uncionar*. Nesse caso, a palavra no idioma alvo "trabalha" é alinhada com "*works*", no idioma pivô, que, entre outros alinhamentos, é alinhada com "funcionar". O contexto onde o par ocorre é apresentado na tabela 4.5. Em outro contexto, o par *trabalhar*<>*funcionar* poderia ser considerado paráfrase.

Tabela 4.5 - Contexto de avaliação do par trabalhar<>funcionar.

A advogada brasileira Natália Santanna, 30, vive nos Estados Unidos desde os 18 anos e ***trabalha*** com direito de imigração. Ela faz defesa voluntária de famílias de imigrantes não documentados, que fogem da violência na América Central e são colocadas em centros de detenção no sul dos EUA.

Em outros casos, devido a uma das palavras do par compor uma expressão multipalavra, o par não foi considerado paráfrase. Isso ocorre, por exemplo, no par *exterior*<>*externo*, apresentado na Tabela 4.6.

Tabela 4.6 - Contexto de avaliação do par exterior<>externo.

São 2.136 páginas de telegramas produzidos com grau de sigilo reservado, mas que tiveram a classificação cancelada e foram divulgados pelo *ministério das relações* ***exteriores*** nesta terça (16), a partir de um requerimento da revista " época " por meio da lei de acesso a informação.

Como pôde ser visto na tabela 4.6, não é possível intercambiar *"exteriores"* por *"externos"*, já que alteraria o significado do termo "*ministério das relações exteriores*".

Outro par em que uma das palavras faz parte de um nome composto é *consumidor<>consumo*. No caso da sentença onde o par ocorreu (apresentada na tabela 4.7), não é possível intercambiar as palavras porque a palavra "*consumido*r" faz parte da expressão multipalavra "*código de defesa do consumidor*".

Tabela 4.7 - Contexto de avaliação do par consumidor<>consumo.

Produtos famosos, como o sorvete Kibon , o sabão em pó Omo , o desodorante Rexona men v 8 , os sorvetes Chocolover e a aveia Quacker estão na lista dos itens em que foi constatado descumprimento do *código de defesa do* ***consumidor*** .

Em outros casos, os juízes avaliaram instâncias como não sendo paráfrases por conta de erros de etiquetação. Um caso em que isso ocorre é com o par *caminhar<>trilhar*. Nesse caso, o substantivo "*caminhada*" (originalmente encontrado no corpus) foi lematizado incorretamente (tornando-se "*caminhar*"), Algo semelhante aconteceu com o substantivo "*trilha",* lematizado como se fosse o verbo "*trilhar*". Nesse caso, o par poderia ser considerado correto se fosse "*caminhada*" e "*trilha*", ou seja, se "*caminhada*" tivesse sido etiquetada corretamente como substantivo. A tabela 4.8 apresenta a sentença onde esse caso ocorre.

Tabela 4.8 - Contexto de avaliação do par caminhar<>trilhar.

Melhor época no inverno quanto entre R$ 400 e R$ 700 por pessoa, dependendo do tamanho do grupo, com guia quem leva consulte a lista de guias certificados no site parnaso.tur.br; por_causa_de incêndios na região, a travessia está fechada e não há previsão para reabertura pacote uma noite em Petrópolis, no hotel abrigo do açú, custa R$ 650. inclui traslado, ida a cachoeira véu da noiva e ***caminhada*** até os castelos do açú. na agência nattrip: (21) 3264 - 0182; nattrip.com br.

Em outro caso com erro de lematização, "*áfrica*" (nome próprio) foi lematizada tornando-se *"áfrico*" (substantivo). Isso possibilitou que o par *áfrico<>africano* pudesse ser gerado. A tabela 4.9 apresenta a sentença onde o caso ocorre.

Tabela 4.9 - Contexto de avaliação do par áfrico<>africano.

" Antes da etapa da Jeffreys Bay , na ***África*** do Sul , ficamos uns quatro dias só fazendo pranchas para ele testar . dia , noite e madrugada " , diz . desconfiança.

Outro erro, esse mais incomum, acontece quando uma das palavras pertencentes ao par de instâncias faz parte de uma sigla. No exemplo *central<>centro*, a palavra central foi encontrada originalmente no corpus na sigla central única dos trabalhadores. Casos desse tipo foram avaliados com "não" pelo fato do intercâmbio das palavras do par afetar um nome. A sentença onde o caso ocorre está expressa na Tabela 4.10.

Tabela 4.10 - Contexto de avaliação do par central<>centro.

O manifesto do ato, divulgado na semana passada por organizações como cut (***central*** única dos trabalhadores), une (união nacional dos estudantes) e mtst, ainda defende a saída de eduardo cunha da presidência da câmara e cita pautas tradicionais da esquerda, como as reformas tributária e agrária.

Esses casos de erro ilustram as limitações e problemas da estratégia de identificação de paráfrases atualmente adotada no NEPaL. Como alternativas para o tratamento desses casos, pode ser utilizado um conjunto de identificadores de expressões multipalavras (VILLAVICENCIO et al., 2010) ou um desambiguador de sentido como em Aziz e Specia (2013).

Capítulo 5

CONCLUSÃO

Este capítulo tem como objetivo apresentar ao leitor as conclusões obtidas durante todo o desenvolvimento do projeto e também as limitações do NEPaL, bem como algumas possibilidades de trabalhos futuros.
Este capítulo está organizado da seguinte forma: na seção 5.1 são apresentadas as conclusões deste trabalho e a seção 5.2 traz algumas propostas de trabalhos futuros.

5.1 Conclusões

A partir dos experimentos realizados neste projeto (veja Capítulo 4) é possível concluir que é viável aplicar a estratégia de AMSF para aprender paráfrases de forma incremental e automática comprovando, assim, a hipótese inicialmente estabelecida. Apesar de terem sido executadas apenas 15 iterações durante os experimentos, agrupadas em três versões do módulo Promotor (gerações), pôde-se notar que as paráfrases aprendidas foram úteis para novos aprendizados, além de que, no decorrer das iterações, a porcentagem de crenças promovidas corretamente só aumentou, como pôde ser visto no gráfico que compõe a Figura 4.2 (Capítulo 4).

Vale ressaltar que o reconhecimento automático de paráfrases é uma área de pesquisa bastante incipiente no Brasil, sendo que até o momento o único trabalho para o idioma português brasileiro do qual se tem conhecimento é o de SENO (2010), baseado em uma abordagem simbólica que utiliza diversos tipos de conhecimento linguístico dependentes de língua. Enquanto isso, a abordagem aqui investigada se baseou em um modelo probabilístico, independente de língua. Tal abordagem, que é baseada no método proposto por Bannard e Callison-Burch

(2005), se mostrou válida quando aplicada ao idioma português do Brasil. Conforme experimentos descritos na Seção 4.1, a precisão média obtida pelo NEPAL durante as 15 iterações é de cerca de 79%. Para o idioma inglês, Bannard e Callison-Burch (2005) relatam uma precisão média de 64,5%, quando utilizado o alinhamento automático e o significado correto das paráfrases no contexto avaliado (assim como no NEPAL). Além disso, também é importante citar a validação do método de Bannard e Callison-Burch (2005) quando aplicado para outro idioma, lembrando que no referido trabalho os autores citam pesquisas para encontrar paráfrases em inglês utilizando o alemão como idioma pivô. Neste trabalho, com a utilização da mesma estratégia, foi possível alinhar paráfrases em português utilizando o inglês como idioma pivô.

É possível afirmar também que os resultados apresentados nesse trabalho podem ser comparados com os resultados apresentados por Bannard e Callison-Burch (2005). Em comparação com Bannard e Callison-Burch (2005), em se tratando apenas de alinhamentos automáticos, assim como acontece no Nepal, os resultados obtidos são melhores: enquanto Bannard e Callison-Burch (2005) atinge precisão de 64,5%, quanto é considerado o significado correto no contexto, o NEPaL obtém, em média, cerca de 79,42% de precisão durante as 15 iterações.

É importante mencionar, também, que existem limitações inerentes à versão atual do sistema. Uma das limitações é que existem casos nos quais candidatas a paráfrases fazem parte de a uma expressão multipalavra. É possível que esse problema possa ser resolvido com a utilização de uma ferramenta de alinhamento diferente da utilizada no projeto, que seja capaz de alinhar sintagmas ou pequenas expressões.

Outro problema ocorre por conta da lematização incorreta guiada por erro de etiquetação. Alguns candidatos a paráfrases foram consideradas errados porque a lematização não foi feita corretamente.

Apesar dessas limitações, pode-se afirmar que a estratégia utilizada é útil para o reconhecimento e a extração de paráfrases em português, atingindo resultados satisfatórios. Desse modo, o NEPaL se apresenta como uma alternativa para o árduo trabalho manual de geração de lista de paráfrases por especialistas.

5.2 Trabalhos Futuros

Uma das possibilidades de trabalho futuro é a extensão do modelo aqui desenvolvido para aprender paráfrases também em nível de sintagmas, como dito inicialmente. Atualmente, a ferramenta utilizada para o alinhamento lexical frequentemente gera alinhamentos do tipo um-para-um, sendo que há pouquíssimos casos de alinhamentos entre grupos de palavras no corpus (ou seja, alinhamentos do tipo um-para-muitos e muitos-para-muitos), o que dificulta o reconhecimento de paráfrases em nível de sintagmas. Para tratar dessa limitação, seria interessante empregar um identificador de expressões multipalavras como o mwetoolkit (RAMISCH, 2015). Como mencionado anteriormente (subseção 1.1), o NEPaL extrai apenas paráfrases lexicais. Com a utilização de um identificador de expressões multipalavras, seria possível também gerar paráfrases em nível de sintagmas, assim como no projeto de Bannard e Callison-Burch (2005). Outra forma de alinhamento muitos-para-muitos seria o uso de *chunkers,* possibilitando o agrupamento prévio de palavras antes do alinhamento. Dessa forma é possível identificar sintagmas nominais.

Outra possibilidade é investigar novas fontes de extração de textos para a formação de corpus. Na fonte atual, o jornal *online* Folha de São Paulo, as noticiais não são publicadas frequentemente, tornando a formação do corpus muito lenta. Além disso, as traduções das notícias não são traduções exatamente iguais. Como foi mencionado na seção 3.1.1, existem sentenças que ocorrem em apenas um dos textos que formam o par de textos paralelos. Isso dificulta o alinhamento, comprometendo todo o processo. Fontes alternativas seriam o corpus FAPESP, além de outras fontes de notícias *online*, como o portal G1.

Outro ponto a ser investigado e que pode trazer melhorias em termos de aprendizado é aumento do corpus. Por se tratar de um modelo probabilístico, acredita-se que com um corpus maior, provavelmente os resultados seriam melhores e haveria uma maior variabilidade das paráfrases.

Para validar os resultados obtidos neste trabalho, seria importante realizar uma avaliação extrínseca por meio do uso das paráfrases extraídas em aplicações como a Tradução Automática e a Sumarização Automática, utilizando os resultados como um recurso. Nesse sentido, diversos protótipos de sistemas desenvolvidos

pelo NILC[34]. Dessa forma, será possível avaliar se a utilização das paráfrases de fato contribui para melhorar o desempenho dessas aplicações.

Visando eliminar problemas de lematização e etiquetação, como os exemplos apresentados na seção 4.3 (*caminhar<>trilhar* e *áfrico<>africano)*, pretende-se também realizar as tarefas de lematização e remoção de *stopwords* no Módulo Pré-Processador para que o contexto seja levado em consideração.

Outra linha de investigação futura está relacionada a novas formas de classificação de instâncias. A atual classificação do NEPaLE aceita apenas as opções SIM e NÃO. Uma opção interessante seria utilizar uma classificação numérica, como notas entre 0 e 10, por exemplo. Isso possibilitaria também investigar outra forma de avaliação de concordância, usando, por exemplo, correlação.

Por fim, vale mencionar que em breve as paráfrases aprendidas pelo NEPaL serão disponibilizadas para toda a comunidade no site do aprendizado sem-fim do LALIC:http://www.lalic.dc.ufscar.br/never-ending/.

[34] Núcleo Interinstitucional de Linguística Computacional. USP – São Carlos.

REFERÊNCIAS

ANDROUTSOPOULOS, I.; MALAKASIOTIS, P. A Survey of Paraphrasing and Textual Entailment Methods. **Journal…** Artificial Intelligence Research 38. p. 135–187, may. 2010.

ALEIXO, P.; PARDO, T. A. S. CSTNews: Um Corpus de Textos Jornalísticos Anotados segundo a Teoria Discursiva Multidocumento CST. **Série de Relatórios Técnicos…** Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos/SP, 15 p. 2008.

AZIZ, W.; SPECIA, L. Fully Automatic Compilation of Portuguese-English and Portuguese-Spanish Parallel Corpora. **Proceedings…** 8th Brazilian Symposium in Information and Human Language Technology (STIL-2011), p. 234-238, Cuiaba, Brazil, oct. 2011.

AZIZ, W.; SPECIA, L. Multilingual WSD-like Constraints for Paraphrase Extraction. **Proceedings…** Seventeenth Conference on Computational Natural Language Learning (CoNLL), Sofia, Bulgaria. p. 202-211, aug. 2013.

BANNARD, C.; CALLISON-BURCH, C. Paraphrasing with Bilingual Parallel Corpora. **Proceeding…** Association for Computational Linguistics - ACL, Ann Arbor, USA, p. 597-604, jun. 2005.

BARZILAY, B.; MCKEOWN, K. Extracting Paraphrases from a Parallel Corpus.: **Proceedings…** Association for Computational Linguistics - ACL, Pittsburg, PA, p. 50-57, jun. 2001.

BARZILAY, R.; ELHADAD, N.; MCKEOWN, K. Inferring strategies for sentence ordering in multidocument news summarization. Journal… Artificial Intelligence Research, v. 17 n. 2, p. 35-55, dec. 2002.

BARZILAY, R.; MCKEOWN, K. Sentence Fusion for Multi-document News Summarization. **Computational Linguistics**, v. 31, n. 3, p. 297-327, sep. 2005.

BICK, E. The Parsing System “Palavras” - Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework, **Thesis**. Aarhus University. Aarhus University Press. Denmark (2000).

BLONDEL V. D.; SENNELART P.: Automatic extraction of synonyms in a dictionary. **Proceedings…** SIAM Workshop on Text Mining. Arlington, USA, p. 7-13, apr 2002.

BOLSHAKOV, I. A.; GELBUKH, A. Synonymous Paraphrasing Using WordNet and Internet. **Proceedings…** NLDB, Salford, UK, p. 312-323, jun. 2004.

BRANTS, T.; FRANZ, A.: Web 1T 5-gram Version 1. 2006.

BROWN, P.; PIETRA, S. D.; PIETRA, V. D.; MERCER, R. The mathematics of machine translation: Parameter estimation. **Computational Linguistics**, v. 19, n. 2, p. 263–311, jun. 1993.

CALLISON-BURCH, C.; KOEHN, P.; OSBORNE, M. Improved statistical machine translation using paraphrases. **Proceedings…** HLT Conference of the NAACL, New York. NY, p. 17–24, jun. 2006.

CARLETTA, J. (1996). Assessing agreement on classification tasks: The kappa statistic. **Computational Linguistics**, Eindhoven, Netherlands, v. 22, n. 2, p. 249-254, nov. 1996.

CASELI, H. Indução de léxicos bilíngues e regras para a tradução automática. Indução de léxicos bilíngues e regras para a tradução automática. 2007. **Tese** (Doutorado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil, abr. 2007.

CHANG, C-C; LIN, C-J.: LIBSVM: A library for support vector machines. **Proceedings…** ACM Transactions on Intelligent Systems and Technology (TIST), v. 2, n. 3, p. 27, apr 2011.

CHURCH, K.; GALE, W.; HANKS, P.; HINDLE, D.: Parsing, word associations and typical predicate-argument relations. **Proceedings…** workshop on Speech and Natural Language. Association for Computational Linguistics, Stroudsburg, USA. p. 75-81, oct. 1989.

GALE, W; CHURCH, K. A program for aligning sentences in bilingual corpora. **Proceedings…** 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, USA, p. 1–8, jun. 1991.

GANITKEVITCH, J.; VAN DURME, B.; CALLISON-BURCH, C.: PPDB: The Paraphrase Database. **Proceedings…** HLT-NAACL, Atlanta, USA. p. 758-764. jun 2013.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1. 2009.

HERRERA, J.; PEÑAS, A.; VERDEJO, F. Textual entailment recognition based on dependency analysis andWordNet. Textual entailment recognition based on dependency analysis and wordnet. **Proceedings…** PASCAL Workshop on Recognizing Textual Entailment, Southampton, UK, p. 21-24, 33–36, apr. 2005.

HSU, W. J.; DU, M. W. New algorithms for the LCS problem. **Journal…** Computer and System Sciences, v. 29, n. 2, p. 133-152, dec 1984.

IBRAHIM, A.; KATZ, B.; LIN, J. Extracting structural paraphrases from aligned monolingual corpora. Proceedings… Second InternationalWorkshop on Paraphrasing (ACL 2003), Sapporo, Japan, p. 57-64, jul. 2003.

IORDANSKAJA, L.; KITTREDGE, R.; POLGUÈRE, A. Lexical Selection and Paraphrase in a Meaning-text Generation Model. Cecile L. Paris, William R. Swartout and William C. Mann (editors), Natural Language Generation… Artificial Intelligence and Computational Linguistics, p. 293-312. Kluwer Academic, Publishers, 1991.

KOEHN, P. Europarl. A multilingual corpus for evaluation of machine translation. **Projeto**, 2002.

KOEHN, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. **Proceedings…** Tenth Machine Translation Summit, Phuket, Thailand, p. 79–86, sep. 2005.

KOEHN, P., HOANG, H., BIRCH, A., CALLISON-BURCH, C., FEDERICO, M., BERTOLDI, N., COWAN, B., SHEN, W., MORAN, C., ZENS, R., DYER, C., BOJAR, O., CONSTANTIN, A., HERBST, E. Moses: open source toolkit for statistical machine translation. **Proceedings…** 45th Annual Meeting of the Association for Computational Linguistics: Demo and Poster Sessions, Prague, Czech Republic, p. 177–180, sep. 2007.

KORFHAGE, Robert R. Information Retrieval and Storage. 1997.

LEWIS, D. D.: Representation and learning in information retrieval. **Ph.D**. Dissertation, Amherst, USA. 1992.

LIN, D.; PANTEL, P.: DIRT@ SBT@ discovery of inference rules from text. In: **Proceedings…** seventh ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, San Francisco, USA, p. 323-328, aug 2001.

LIN, D.; CHURCH, K.; JI, H.; SEKINE, S.; YAROWSKY, D.; BERGSMA, S.; PATIL, K.; PITLER, E.; LATHBURY, R.; RAO, V.; DALWANI, K.; NARSALE, S.: New tools for web-scale n-grams. Proceedings… LREC. Gozo and Comino, Malta, p. 221–2227, may 2010.

MALAKASIOTIS, P. Paraphrase recognition using machine learning to combine similarity measures. **Proceedings…** ACL-IJCNLP 2009 Student Research Workshop. Association for Computational Linguistics, Suntec, Singapore . p. 27-35, aug 2009.

MANI, I. Automatic Summarization. **John Benjamins** Publishing Co. Amsterdam, Netherlands, 2001, 297 p.

MAZIERO, E. G.; PARDO, T. A. S.; Di FELIPPO, A.; DIAS-DA-SILVA, B. C. A Base de Dados Lexical e a Interface Web do TeP 2,0 - Thesaurus Eletrônico para o Português do Brasil. **Anais**… VI Workshop Tecnologia da Informação e da Linguagem Humana (TIL), Vila Velha, ES, p. 390-392, out 2008.

MIKHEEV, A. LT POS - The LTG part of speech tagger. **Language Technology Group**. University of Edinburgh, 1997.

MILLER, G. A.; BECKWITH, R.; FELLBAUM, C.; GROSS, D.; MILLER, K. J. Introduction to WordNet: An on-line lexical database. **International Journal…** Lexicography (special issue), Oxford Univ Press, v. 3, n. 4, p. 235–245, 1990.

MITCHELL, T. M.; BETTERIDGE, J.; CARLSON, A.; HONG, S. A.; HRUSCKA, E. a. L.-M. E.;WANG, S. Never-ending language learning: **The readtheweb manifesto**. In: [S.l.: s.n.], 2008.

NAPOLES, C.; GORMLEY, M.; VAN DURME, B.: Annotated gigaword. **Proceedings …** Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction. Association for Computational Linguistics, Montreal, Canada. p. 95-100, jun 2012.

OCH, F. J.; NEY, H. A systematic comparison of various statistical alignment models. **Computational Linguistics**. v. 29, n. 1, p. 19–51, mar. 2003.

PANG, B.; KNIGHT, K.; MARCU, D. Syntax-based Alignment of Multiple Translations: Extracting Paraphrases and Generating New Sentences. Edmonton, Canada, **Proceedings…** Human Language Technology Conference – HLT/NAACL, p. 102-109, may. 2003.

QUINLAN, J. R.: C4.5: Programs for machine learning. **Morgan Kaufmann** Publishers Inc., San Francisco, USA, 1993.

QUIRK, C.; BROCKETT, C.; DOLAN, W. B. Monolingual machine translation for paraphrase generation. **Proceedings…** Conference on EMNLP, Barcelona, Spain. p. 142–149, jul. 2004.

RAMISCH, C. Multiword Expressions Acquisition: A Generic and Open Framework", Theory and Applications of Natural Language Processing series XIV, Springer, ISBN 978-3-319-09206-5, 230 p., 2015.

ROSE, T.G.; STEVENSON, M.; WHITEHEAD, M. "The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources". **Proceedings…** Third International Conference on Language Resources and Evaluation, Las Palmas de Gran Canaria, Spain, p. 29-31, may 2002.

SENO, E. R. M. Fusão de sentenças similares em português para o tratamento de redundância na Sumarização Multidocumento. **Qualificação…** Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil, 58 p., feb. 2007.

SENO, E. R. M. NUNES, M.G.V. Reconhecimento de Informações Comuns para a Fusão de Sentenças Comparáveis do Português. In: **Revista Linguamática**, nº 1, p.71-87. jan. 2009.

SENO, E. R. M.: Um método para a fusao automática de sentencas similares em português. **Tese de Doutorado**. Universidade de São Paulo. 2010.

SHIMOHATA, M.; SUMITA, E. Identifying synonymous expressions from a bilingual corpus for example-based machine translation. Proceedings… 19th In-

ternational Conference on Computational Linguistics (COLING) Workshop on Machine Translation in Asia. Stroudsburg, USA . p. 1-6. Sep, 2002.

SILVA, J. W. F. Aquisição de Conhecimento de Mundo para Sistema de Processamento de Linguagem Natural – **Dissertação** (Mestrado) – Universidade Federal do Ceará, Fortaleza, Brasil, 2013.

SZPEKTOR, I.; TANEV, H.; DAGAN, I.; COPPOLA, B. Scaling web-based acquisition of entailment relations. **Proceedings…** EMNLP, Barcelona, Spain. p. 41–8, jul. 2004.

TEIXEIRA, R. O.; SENO, E. R. M.; CASELI, H. M. NePaLE: Uma ferramenta computacional de suporte à avaliação de paráfrases. **Proceedings…** IV Workshop de Iniciação Científica em Tecnologia da Informação e da Linguagem Humana, Natal, Brazil. p. 1-4, nov, 2015.

VAN DER PLAS, L.; TIEDEMANN, J. Finding synonyms using automatic word alignment and measures of distributional similarity. **Proceedings…** COLING/ACL on Main conference poster sessions. Association for Computational Linguistics, Sydney, Australia. p. 866-873, jul. 2006.

VIEIRA, T. L. Aprendizado Sem-Fim de Equivalentes Lexicais Bilíngues. **Qualificação…** Dissertação (Mestrado) — Universidade Federal de São Carlos, São Carlos, Brasil. apr. 2013.

VIEIRA, T. L.; CASELI, H. M.: PorTAl: Recursos e Ferramentas de Tradução Automática para o Português do Brasil. **Proceedings…** 8th Brazilian Symposium

in Information and Human Language Technology (STIL), Cuiabá, Brazil, p. 179-183. oct 2011.

VOSSEN, P.: Introduction to eurowordnet. EuroWordNet: A multilingual database with lexical semantic networks. Springer Netherlands, p. 1-17, 1998.

WU, H.; ZHOU, M.: Optimizing synonym extraction using monolingual and bilingual resources. **Proceedings...** second international workshop on Paraphrasing-Volume 16. Association for Computational Linguistics, Stroudsburg, USA, p. 72-79, jul 2003.

ZHAO, S.; WANG, H.; LAN, X.; LIU, T.: (2010). Leveraging multiple mt engines for paraphrase generation. **Proceedings…** 23rd International Conference on Computational Linguistics (COLING), Beijing, China, p. 1326–1334, aug. 2010.

Apêndice A

ÁRVORES DE DECISÃO

Nesse apêndice são apresentadas as árvores de decisão geradas pelo aprendizado das versões do Promotor utilizando o algoritmo de J48.

Árvore gerada durante treinamento do Promotor-0:

Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree

```
probabilidade <= 9.475
|  probabilidade <= 6.198
|  |  LCSR <= 0.091
|  |  |  LCSR <= 0: no (10.0)
|  |  |  LCSR > 0
|  |  |  |  LCSR <= 0.077
|  |  |  |  |  probabilidade <= 0.82: yes (5.0)
|  |  |  |  |  probabilidade > 0.82
|  |  |  |  |  |  probabilidade <= 0.84: no (5.0)
|  |  |  |  |  |  probabilidade > 0.84
|  |  |  |  |  |  |  probabilidade <= 0.871: yes (5.0)
|  |  |  |  |  |  |  probabilidade > 0.871: no (5.0)
|  |  |  |  LCSR > 0.077
|  |  |  |  |  LCSR <= 0.083: no (15.0)
|  |  |  |  |  LCSR > 0.083
|  |  |  |  |  |  probabilidade <= 0.669: yes (5.0)
|  |  |  |  |  |  probabilidade > 0.669
|  |  |  |  |  |  |  probabilidade <= 0.735: no (15.0)
|  |  |  |  |  |  |  probabilidade > 0.735
|  |  |  |  |  |  |  |  probabilidade <= 0.76: yes (5.0)
|  |  |  |  |  |  |  |  probabilidade > 0.76
|  |  |  |  |  |  |  |  |  probabilidade <= 2.145: no (10.0)
|  |  |  |  |  |  |  |  |  probabilidade > 2.145
|  |  |  |  |  |  |  |  |  |  probabilidade <= 3.704: yes (5.0)
|  |  |  |  |  |  |  |  |  |  probabilidade > 3.704: no (5.0)
|  |  LCSR > 0.091
|  |  |  LCSR <= 0.1
|  |  |  |  probabilidade <= 0.852: no (5.0)
|  |  |  |  probabilidade > 0.852: yes (50.0)
|  |  |  LCSR > 0.1
|  |  |  |  probabilidade <= 0.968
```

```
| | | | | LCSR <= 0.125
| | | | | | probabilidade <= 0.55: yes (5.0)
| | | | | | probabilidade > 0.55: no (15.0)
| | | | | LCSR > 0.125
| | | | | | probabilidade <= 0.809
| | | | | | | probabilidade <= 0.563: yes (30.0)
| | | | | | | probabilidade > 0.563
| | | | | | | | probabilidade <= 0.565: no (5.0)
| | | | | | | | probabilidade > 0.565
| | | | | | | | | probabilidade <= 0.793
| | | | | | | | | | LCSR <= 0.2
| | | | | | | | | | | LCSR <= 0.154: yes (5.0)
| | | | | | | | | | | LCSR > 0.154
| | | | | | | | | | | | probabilidade <= 0.672: no (10.0)
| | | | | | | | | | | | probabilidade > 0.672: yes (5.0)
| | | | | | | | | | LCSR > 0.2: yes (35.0)
| | | | | | | | | probabilidade > 0.793: no (5.0)
| | | | | | | probabilidade > 0.809: yes (55.0)
| | | | probabilidade > 0.968
| | | | | LCSR <= 0.545
| | | | | | probabilidade <= 1.398
| | | | | | | LCSR <= 0.417
| | | | | | | | probabilidade <= 1.025: no (25.0)
| | | | | | | | probabilidade > 1.025
| | | | | | | | | probabilidade <= 1.072: yes (20.0)
| | | | | | | | | probabilidade > 1.072
| | | | | | | | | | LCSR <= 0.182
| | | | | | | | | | | probabilidade <= 1.338
| | | | | | | | | | | | probabilidade <= 1.141: yes (5.0)
| | | | | | | | | | | | probabilidade > 1.141: no (15.0)
| | | | | | | | | | | probabilidade > 1.338: yes (10.0)
| | | | | | | | | | LCSR > 0.182: no (35.0)
| | | | | | | LCSR > 0.417: yes (20.0)
| | | | | | probabilidade > 1.398
| | | | | | | probabilidade <= 2.857
| | | | | | | | LCSR <= 0.154
| | | | | | | | | probabilidade <= 1.784
| | | | | | | | | | probabilidade <= 1.507: no (5.0)
| | | | | | | | | | probabilidade > 1.507
| | | | | | | | | | | probabilidade <= 1.563: yes (15.0)
| | | | | | | | | | | probabilidade > 1.563
| | | | | | | | | | | | probabilidade <= 1.612: no (5.0)
| | | | | | | | | | | | probabilidade > 1.612: yes (15.0)
| | | | | | | | | probabilidade > 1.784
| | | | | | | | | | probabilidade <= 2.461: no (15.0)
| | | | | | | | | | probabilidade > 2.461: yes (5.0)
| | | | | | | | LCSR > 0.154
| | | | | | | | | probabilidade <= 1.998
| | | | | | | | | | probabilidade <= 1.675: yes (40.0)
| | | | | | | | | | probabilidade > 1.675
| | | | | | | | | | | probabilidade <= 1.714: no (10.0)
| | | | | | | | | | probabilidade > 1.714
| | | | | | | | | | | LCSR <= 0.455
| | | | | | | | | | | | probabilidade <= 1.997
| | | | | | | | | | | | | LCSR <= 0.2
| | | | | | | | | | | | | | LCSR <= 0.182: yes (5.0)
| | | | | | | | | | | | | | LCSR > 0.182: no (5.0)
| | | | | | | | | | | | | LCSR > 0.2: yes (30.0)
| | | | | | | | | | | | probabilidade > 1.997: no (5.0)
| | | | | | | | | | | LCSR > 0.455: no (5.0)
| | | | | | | | | probabilidade > 1.998: yes (60.0)
| | | | | | | probabilidade > 2.857
| | | | | | | | probabilidade <= 3.199: no (15.0)
| | | | | | | | probabilidade > 3.199
| | | | | | | | | probabilidade <= 4.373
```

```
| | | | | | | | | | probabilidade <= 3.741
| | | | | | | | | | | LCSR <= 0.167: yes (15.0)
| | | | | | | | | | | LCSR > 0.167
| | | | | | | | | | | | probabilidade <= 3.299: yes (5.0)
| | | | | | | | | | | | probabilidade > 3.299
| | | | | | | | | | | | | probabilidade <= 3.409: no (10.0)
| | | | | | | | | | | | | probabilidade > 3.409
| | | | | | | | | | | | | | LCSR <= 0.222: no (5.0)
| | | | | | | | | | | | | | LCSR > 0.222: yes (15.0)
| | | | | | | | | | probabilidade > 3.741
| | | | | | | | | | | LCSR <= 0.2: no (25.0)
| | | | | | | | | | | LCSR > 0.2
| | | | | | | | | | | | LCSR <= 0.3
| | | | | | | | | | | | | probabilidade <= 4.024: yes (5.0)
| | | | | | | | | | | | | probabilidade > 4.024: no (5.0)
| | | | | | | | | | | | LCSR > 0.3: no (5.0)
| | | | | | | | | probabilidade > 4.373
| | | | | | | | | | LCSR <= 0.385
| | | | | | | | | | | probabilidade <= 4.539: yes (15.0)
| | | | | | | | | | | probabilidade > 4.539
| | | | | | | | | | | | probabilidade <= 4.698: no (10.0)
| | | | | | | | | | | | probabilidade > 4.698
| | | | | | | | | | | | | LCSR <= 0.111: no (5.0)
| | | | | | | | | | | | | LCSR > 0.111
| | | | | | | | | | | | | | LCSR <= 0.2
| | | | | | | | | | | | | | | LCSR <= 0.182: yes (25.0)
| | | | | | | | | | | | | | | LCSR > 0.182: no (10.0)
| | | | | | | | | | | | | | LCSR > 0.2: yes (30.0)
| | | | | | | | | LCSR > 0.385: no (5.0)
| | | | | LCSR > 0.545
| | | | | | LCSR <= 0.667: no (25.0)
| | | | | | LCSR > 0.667: yes (5.0)
| probabilidade > 6.198: yes (95.0/5.0)
probabilidade > 9.475
| probabilidade <= 14.049
| | probabilidade <= 12.94
| | | LCSR <= 0.333
| | | | probabilidade <= 10.949
| | | | | probabilidade <= 10.727
| | | | | | probabilidade <= 10.041: no (20.0)
| | | | | | probabilidade > 10.041
| | | | | | | LCSR <= 0.083: no (15.0)
| | | | | | | LCSR > 0.083
| | | | | | | | LCSR <= 0.091: yes (5.0)
| | | | | | | | LCSR > 0.091
| | | | | | | | | probabilidade <= 10.589
| | | | | | | | | | probabilidade <= 10.257
| | | | | | | | | | | probabilidade <= 10.136: yes (10.0)
| | | | | | | | | | | probabilidade > 10.136: no (20.0)
| | | | | | | | | | probabilidade > 10.257
| | | | | | | | | | | probabilidade <= 10.266: yes (5.0)
| | | | | | | | | | | probabilidade > 10.266
| | | | | | | | | | | | probabilidade <= 10.292: no (5.0)
| | | | | | | | | | | | probabilidade > 10.292
| | | | | | | | | | | | | probabilidade <= 10.455
| | | | | | | | | | | | | | LCSR <= 0.143: yes (15.0)
| | | | | | | | | | | | | | LCSR > 0.143
| | | | | | | | | | | | | | | LCSR <= 0.2
| | | | | | | | | | | | | | | | probabilidade <= 10.42: no (10.0)
| | | | | | | | | | | | | | | | probabilidade > 10.42: yes (5.0)
| | | | | | | | | | | | | | | LCSR > 0.2: yes (5.0)
| | | | | | | | | | | | | probabilidade > 10.455
| | | | | | | | | | | | | | probabilidade <= 10.57: no (10.0)
| | | | | | | | | | | | | | probabilidade > 10.57: yes (10.0)
| | | | | | | | | probabilidade > 10.589: no (25.0)
```

```
| | | | | probabilidade > 10.727
| | | | | | LCSR <= 0.1
| | | | | | | probabilidade <= 10.767: yes (5.0)
| | | | | | | probabilidade > 10.767: no (10.0)
| | | | | | LCSR > 0.1
| | | | | | | probabilidade <= 10.885: yes (25.0)
| | | | | | | probabilidade > 10.885
| | | | | | | | probabilidade <= 10.909: no (5.0)
| | | | | | | | probabilidade > 10.909: yes (15.0)
| | | | probabilidade > 10.949
| | | | | probabilidade <= 12.089
| | | | | | probabilidade <= 11.702
| | | | | | | probabilidade <= 11.12: no (30.0)
| | | | | | | probabilidade > 11.12
| | | | | | | | LCSR <= 0.267
| | | | | | | | | probabilidade <= 11.548
| | | | | | | | | | probabilidade <= 11.537
| | | | | | | | | | | LCSR <= 0.154: no (30.0)
| | | | | | | | | | | LCSR > 0.154
| | | | | | | | | | | | probabilidade <= 11.275: yes (15.0)
| | | | | | | | | | | | probabilidade > 11.275: no (10.0)
| | | | | | | | | | probabilidade > 11.537: yes (10.0)
| | | | | | | | | probabilidade > 11.548: no (25.0)
| | | | | | | | LCSR > 0.267
| | | | | | | | | probabilidade <= 11.275: yes (5.0)
| | | | | | | | | probabilidade > 11.275
| | | | | | | | | | probabilidade <= 11.637: no (10.0)
| | | | | | | | | | probabilidade > 11.637: yes (5.0)
| | | | | | probabilidade > 11.702: no (60.0)
| | | | | probabilidade > 12.089
| | | | | | probabilidade <= 12.838
| | | | | | | probabilidade <= 12.66
| | | | | | | | probabilidade <= 12.539
| | | | | | | | | probabilidade <= 12.479
| | | | | | | | | | LCSR <= 0.167
| | | | | | | | | | | LCSR <= 0.1: yes (5.0)
| | | | | | | | | | | LCSR > 0.1
| | | | | | | | | | | | probabilidade <= 12.09: yes (5.0)
| | | | | | | | | | | | probabilidade > 12.09
| | | | | | | | | | | | | probabilidade <= 12.285: no (25.0)
| | | | | | | | | | | | | probabilidade > 12.285
| | | | | | | | | | | | | | LCSR <= 0.125: no (5.0)
| | | | | | | | | | | | | | LCSR > 0.125: yes (5.0)
| | | | | | | | | | LCSR > 0.167: yes (15.0)
| | | | | | | | | probabilidade > 12.479
| | | | | | | | | | LCSR <= 0.222
| | | | | | | | | | | LCSR <= 0.143: no (5.0)
| | | | | | | | | | | LCSR > 0.143
| | | | | | | | | | | | probabilidade <= 12.517: yes (5.0)
| | | | | | | | | | | | probabilidade > 12.517: no (5.0)
| | | | | | | | | | LCSR > 0.222: no (20.0)
| | | | | | | | probabilidade > 12.539: yes (10.0)
| | | | | | | probabilidade > 12.66: no (25.0)
| | | | | | probabilidade > 12.838: yes (15.0)
| | | LCSR > 0.333
| | | | LCSR <= 0.8
| | | | | probabilidade <= 12.267: yes (20.0)
| | | | | probabilidade > 12.267
| | | | | | LCSR <= 0.385: yes (5.0)
| | | | | | LCSR > 0.385: no (10.0)
| | | | LCSR > 0.8: no (5.0)
| | probabilidade > 12.94
| | | probabilidade <= 13.244: no (55.0)
| | | probabilidade > 13.244
| | | | LCSR <= 0.182
```

```
| | | | | probabilidade <= 13.246: yes (5.0)
| | | | | probabilidade > 13.246
| | | | | | probabilidade <= 13.817: no (75.0)
| | | | | | probabilidade > 13.817
| | | | | | | LCSR <= 0.111: yes (5.0)
| | | | | | | LCSR > 0.111: no (25.0)
| | | | LCSR > 0.182
| | | | | LCSR <= 0.2
| | | | | | probabilidade <= 13.428: no (5.0)
| | | | | | probabilidade > 13.428
| | | | | | | probabilidade <= 13.576: yes (5.0)
| | | | | | | probabilidade > 13.576
| | | | | | | | probabilidade <= 13.684: no (5.0)
| | | | | | | | probabilidade > 13.684: yes (5.0)
| | | | | LCSR > 0.2
| | | | | | probabilidade <= 13.602
| | | | | | | LCSR <= 0.222: yes (5.0)
| | | | | | | LCSR > 0.222
| | | | | | | | probabilidade <= 13.294: no (5.0)
| | | | | | | | probabilidade > 13.294
| | | | | | | | | probabilidade <= 13.354: yes (5.0)
| | | | | | | | | probabilidade > 13.354
| | | | | | | | | | probabilidade <= 13.38: no (5.0)
| | | | | | | | | | probabilidade > 13.38
| | | | | | | | | | | probabilidade <= 13.428: yes (5.0)
| | | | | | | | | | | probabilidade > 13.428: no (10.0)
| | | | | probabilidade > 13.602: no (35.0)
| probabilidade > 14.049
| | probabilidade <= 14.201: yes (25.0)
| | probabilidade > 14.201
| | | probabilidade <= 14.57: no (35.0)
| | | probabilidade > 14.57
| | | | LCSR <= 0.571
| | | | | probabilidade <= 64.946
| | | | | | LCSR <= 0.2: no (1215.0/455.0)
| | | | | | LCSR > 0.2
| | | | | | | LCSR <= 0.25
| | | | | | | | LCSR <= 0.222
| | | | | | | | | probabilidade <= 15.973: no (10.0)
| | | | | | | | | probabilidade > 15.973
| | | | | | | | | | probabilidade <= 55.714
| | | | | | | | | | | probabilidade <= 46.439
| | | | | | | | | | | | probabilidade <= 21.429
| | | | | | | | | | | | | probabilidade <= 18.824
| | | | | | | | | | | | | | probabilidade <= 17.442: yes (5.0)
| | | | | | | | | | | | | | probabilidade > 17.442: no (5.0)
| | | | | | | | | | | | | probabilidade > 18.824: yes (15.0)
| | | | | | | | | | | | probabilidade > 21.429
| | | | | | | | | | | | | probabilidade <= 31.269: no (20.0)
| | | | | | | | | | | | | probabilidade > 31.269
| | | | | | | | | | | | | | probabilidade <= 39.189: yes (20.0)
| | | | | | | | | | | | | | probabilidade > 39.189
| | | | | | | | | | | | | | | probabilidade <= 40: no (5.0)
| | | | | | | | | | | | | | | probabilidade > 40
| | | | | | | | | | | | | | | | probabilidade <= 42.295: yes (5.0)
| | | | | | | | | | | | | | | | probabilidade > 42.295: no (5.0)
| | | | | | | | | | | probabilidade > 46.439: yes (25.0)
| | | | | | | | | | probabilidade > 55.714: no (10.0)
| | | | | | | | | LCSR > 0.222
| | | | | | | | | | probabilidade <= 25.127
| | | | | | | | | | | probabilidade <= 24.742
| | | | | | | | | | | probabilidade <= 24.603
| | | | | | | | | | | | probabilidade <= 22.787
| | | | | | | | | | | | | probabilidade <= 21.636
| | | | | | | | | | | | | | probabilidade <= 19.877
```

```
| | | | | | | | | | | | | | | probabilidade <= 19.256
| | | | | | | | | | | | | | | | probabilidade <= 16.828
| | | | | | | | | | | | | | | | | probabilidade <= 16.056: yes (5.0)
| | | | | | | | | | | | | | | | | probabilidade > 16.056: no (5.0)
| | | | | | | | | | | | | | | | probabilidade > 16.828: yes (15.0)
| | | | | | | | | | | | | | | probabilidade > 19.256: no (5.0)
| | | | | | | | | | | | | | | probabilidade > 19.877: yes (10.0)
| | | | | | | | | | | | | | | probabilidade > 21.636: no (5.0)
| | | | | | | | | | | | | | | probabilidade > 22.787: yes (15.0)
| | | | | | | | | | | | | | | probabilidade > 24.603: no (5.0)
| | | | | | | | | | | | | | | probabilidade > 24.742: yes (15.0)
| | | | | | | | | | | | | | probabilidade > 25.127
| | | | | | | | | | | | | | | probabilidade <= 60.694
| | | | | | | | | | | | | | | | probabilidade <= 54.092
| | | | | | | | | | | | | | | | | probabilidade <= 43.836
| | | | | | | | | | | | | | | | | | probabilidade <= 37.487
| | | | | | | | | | | | | | | | | | | probabilidade <= 35.926
| | | | | | | | | | | | | | | | | | | | probabilidade <= 26.743: no (5.0)
| | | | | | | | | | | | | | | | | | | | probabilidade > 26.743
| | | | | | | | | | | | | | | | | | | | | probabilidade <= 35.013
| | | | | | | | | | | | | | | | | | | | | | probabilidade <= 28.571: yes (5.0)
| | | | | | | | | | | | | | | | | | | | | | probabilidade > 28.571
| | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 28.849: no (5.0)
| | | | | | | | | | | | | | | | | | | | | | | probabilidade > 28.849
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 29.159: yes (5.0)
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 29.159
| | | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 29.677: no (5.0)
| | | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 29.677
| | | | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 31.048: yes (10.0)
| | | | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 31.048
| | | | | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 32.712: no (5.0)
| | | | | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 32.712: yes (5.0)
| | | | | | | | | | | | | | | | | | | probabilidade > 35.013: no (5.0)
| | | | | | | | | | | | | | | | probabilidade > 35.926: yes (10.0)
| | | | | | | | | | | | | | | probabilidade > 37.487: no (30.0)
| | | | | | | | | | | | | | probabilidade > 43.836
| | | | | | | | | | | | | | | probabilidade <= 52.174: yes (20.0)
| | | | | | | | | | | | | | | probabilidade > 52.174
| | | | | | | | | | | | | | | | probabilidade <= 53.107: no (5.0)
| | | | | | | | | | | | | | | | probabilidade > 53.107: yes (5.0)
| | | | | | | | | | | | | | probabilidade > 54.092: no (10.0)
| | | | | | | | | | | | | | | probabilidade > 60.694: yes (10.0)
| | | | | | | | | | | | | LCSR > 0.25
| | | | | | | | | | | | | | probabilidade <= 15.805
| | | | | | | | | | | | | | | probabilidade <= 15.187
| | | | | | | | | | | | | | | | probabilidade <= 15.054: yes (10.0)
| | | | | | | | | | | | | | | | probabilidade > 15.054: no (5.0)
| | | | | | | | | | | | | | | probabilidade > 15.187: yes (15.0)
| | | | | | | | | | | | | | probabilidade > 15.805
| | | | | | | | | | | | | | | LCSR <= 0.333
| | | | | | | | | | | | | | | probabilidade <= 41.63
| | | | | | | | | | | | | | | | probabilidade <= 39.417
| | | | | | | | | | | | | | | | | probabilidade <= 33.128
| | | | | | | | | | | | | | | | | | probabilidade <= 28.29: no (135.0/40.0)
| | | | | | | | | | | | | | | | | | probabilidade > 28.29: yes (10.0)
| | | | | | | | | | | | | | | | | probabilidade > 33.128: no (35.0)
| | | | | | | | | | | | | | | | probabilidade > 39.417: yes (10.0)
| | | | | | | | | | | | | | | probabilidade > 41.63
| | | | | | | | | | | | | | | | probabilidade <= 61.483: no (45.0)
| | | | | | | | | | | | | | | | probabilidade > 61.483
| | | | | | | | | | | | | | | | | LCSR <= 0.286: no (5.0)
| | | | | | | | | | | | | | | | | LCSR > 0.286: yes (5.0)
| | | | | | | | | | | | | | | LCSR > 0.333
| | | | | | | | | | | | | | | | probabilidade <= 21.798
| | | | | | | | | | | | | | | | | probabilidade <= 17.442
```

```
| | | | | | | | | | | | probabilidade <= 16.814
| | | | | | | | | | | | | LCSR <= 0.417: no (15.0)
| | | | | | | | | | | | | LCSR > 0.417: yes (5.0)
| | | | | | | | | | | | probabilidade > 16.814: yes (10.0)
| | | | | | | | | | | probabilidade > 17.442: no (45.0)
| | | | | | | | | | probabilidade > 21.798
| | | | | | | | | | | probabilidade <= 22.444: yes (15.0)
| | | | | | | | | | | probabilidade > 22.444
| | | | | | | | | | | | probabilidade <= 39.432
| | | | | | | | | | | | | LCSR <= 0.545
| | | | | | | | | | | | | | LCSR <= 0.5
| | | | | | | | | | | | | | | probabilidade <= 27.778
| | | | | | | | | | | | | | | | probabilidade <= 23.558
| | | | | | | | | | | | | | | | | probabilidade <= 22.622: no (5.0)
| | | | | | | | | | | | | | | | | probabilidade > 22.622: yes (10.0)
| | | | | | | | | | | | | | | | probabilidade > 23.558: no (20.0)
| | | | | | | | | | | | | | | probabilidade > 27.778
| | | | | | | | | | | | | | | | probabilidade <= 35.926: yes (10.0)
| | | | | | | | | | | | | | | | probabilidade > 35.926
| | | | | | | | | | | | | | | | | LCSR <= 0.4
| | | | | | | | | | | | | | | | | | probabilidade <= 37.969: yes (5.0)
| | | | | | | | | | | | | | | | | | probabilidade > 37.969: no (5.0)
| | | | | | | | | | | | | | | | | LCSR > 0.4: no (5.0)
| | | | | | | | | | | | | | | LCSR > 0.5: yes (5.0)
| | | | | | | | | | | | | | LCSR > 0.545: no (5.0)
| | | | | | | | | | | | probabilidade > 39.432
| | | | | | | | | | | | | probabilidade <= 61.782: yes (20.0)
| | | | | | | | | | | | | probabilidade > 61.782: no (5.0)
| | | | | probabilidade > 64.946
| | | | | | LCSR <= 0.214: yes (30.0)
| | | | | | LCSR > 0.214
| | | | | | | LCSR <= 0.286: no (25.0)
| | | | | | | LCSR > 0.286
| | | | | | | | LCSR <= 0.444: yes (20.0)
| | | | | | | | LCSR > 0.444: no (5.0)
| | | | LCSR > 0.571
| | | | | LCSR <= 0.625: no (30.0)
| | | | | LCSR > 0.625
| | | | | | LCSR <= 0.636: yes (10.0)
| | | | | | LCSR > 0.636
| | | | | | | probabilidade <= 23.558: yes (5.0)
| | | | | | | probabilidade > 23.558
| | | | | | | | LCSR <= 0.75: no (20.0)
| | | | | | | | LCSR > 0.75
| | | | | | | | | LCSR <= 0.8: yes (5.0)
| | | | | | | | | LCSR > 0.8: no (10.0)
```

Árvore gerada durante treinamento do Promotor-1:

```
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

probabilidade <= 9.475
|  probabilidade <= 6.198
|  |  LCSR <= 0.091
|  |  |  LCSR <= 0: no (12.0)
|  |  |  LCSR > 0
|  |  |  |  LCSR <= 0.077
|  |  |  |  |  probabilidade <= 0.82: yes (6.0)
|  |  |  |  |  probabilidade > 0.82
|  |  |  |  |  |  probabilidade <= 0.84: no (6.0)
```

```
| | | | | | probabilidade > 0.84
| | | | | | | probabilidade <= 0.871: yes (6.0)
| | | | | | | probabilidade > 0.871: no (6.0)
| | | | LCSR > 0.077
| | | | | LCSR <= 0.083: no (18.0)
| | | | | LCSR > 0.083
| | | | | | probabilidade <= 0.669: yes (6.0)
| | | | | | probabilidade > 0.669
| | | | | | | probabilidade <= 0.735: no (18.0)
| | | | | | | probabilidade > 0.735
| | | | | | | | probabilidade <= 0.76: yes (6.0)
| | | | | | | | probabilidade > 0.76
| | | | | | | | | probabilidade <= 2.145: no (12.0)
| | | | | | | | | probabilidade > 2.145
| | | | | | | | | | probabilidade <= 3.704: yes (6.0)
| | | | | | | | | | probabilidade > 3.704: no (6.0)
| | LCSR > 0.091
| | | probabilidade <= 0.56: yes (30.0)
| | | probabilidade > 0.56
| | | | LCSR <= 0.1
| | | | | probabilidade <= 0.852: no (6.0)
| | | | | probabilidade > 0.852: yes (60.0)
| | | | LCSR > 0.1
| | | | | LCSR <= 0.154
| | | | | | probabilidade <= 1.338
| | | | | | | LCSR <= 0.125: no (30.0)
| | | | | | | LCSR > 0.125
| | | | | | | | probabilidade <= 1.141: yes (18.0)
| | | | | | | | probabilidade > 1.141: no (12.0)
| | | | | | probabilidade > 1.338
| | | | | | | probabilidade <= 1.784
| | | | | | | | probabilidade <= 1.612
| | | | | | | | | probabilidade <= 1.563
| | | | | | | | | | probabilidade <= 1.507
| | | | | | | | | | | probabilidade <= 1.409: yes (6.0)
| | | | | | | | | | | probabilidade > 1.409: no (6.0)
| | | | | | | | | | probabilidade > 1.507: yes (18.0)
| | | | | | | | | probabilidade > 1.563: no (6.0)
| | | | | | | | probabilidade > 1.612: yes (18.0)
| | | | | | | probabilidade > 1.784
| | | | | | | | probabilidade <= 2.461: no (18.0)
| | | | | | | | probabilidade > 2.461
| | | | | | | | | probabilidade <= 3.741
| | | | | | | | | | LCSR <= 0.125: yes (12.0)
| | | | | | | | | | LCSR > 0.125
| | | | | | | | | | | probabilidade <= 3.299: no (6.0)
| | | | | | | | | | | probabilidade > 3.299: yes (6.0)
| | | | | | | | | probabilidade > 3.741
| | | | | | | | | | probabilidade <= 4.373: no (18.0)
| | | | | | | | | | probabilidade > 4.373
| | | | | | | | | | | probabilidade <= 4.539: yes (12.0)
| | | | | | | | | | | probabilidade > 4.539
| | | | | | | | | | | | LCSR <= 0.111: no (6.0)
| | | | | | | | | | | | LCSR > 0.111
| | | | | | | | | | | | | probabilidade <= 4.846: no (6.0)
| | | | | | | | | | | | | probabilidade > 4.846: yes (6.0)
| | | | | LCSR > 0.154
| | | | | | probabilidade <= 0.968
| | | | | | | probabilidade <= 0.809
| | | | | | | | probabilidade <= 0.793
| | | | | | | | | LCSR <= 0.267
| | | | | | | | | | probabilidade <= 0.565: no (6.0)
| | | | | | | | | | probabilidade > 0.565
| | | | | | | | | | | LCSR <= 0.2
| | | | | | | | | | | | probabilidade <= 0.672: no (12.0)
```

```
| | | | | | | | | | | | probabilidade > 0.672: yes (6.0)
| | | | | | | | | | | LCSR > 0.2: yes (24.0)
| | | | | | | | | | LCSR > 0.267: yes (30.0)
| | | | | | | | | probabilidade > 0.793: no (6.0)
| | | | | | | | probabilidade > 0.809: yes (60.0)
| | | | | | | probabilidade > 0.968
| | | | | | | | LCSR <= 0.545
| | | | | | | | | probabilidade <= 1.398
| | | | | | | | | | LCSR <= 0.417
| | | | | | | | | | | LCSR <= 0.182
| | | | | | | | | | | | probabilidade <= 0.998: no (6.0)
| | | | | | | | | | | | probabilidade > 0.998: yes (18.0)
| | | | | | | | | | | LCSR > 0.182
| | | | | | | | | | | | probabilidade <= 1.072
| | | | | | | | | | | | | probabilidade <= 1.025: no (18.0)
| | | | | | | | | | | | | probabilidade > 1.025: yes (12.0)
| | | | | | | | | | | | probabilidade > 1.072: no (42.0)
| | | | | | | | | | LCSR > 0.417: yes (24.0)
| | | | | | | | | probabilidade > 1.398
| | | | | | | | | | probabilidade <= 1.675: yes (48.0)
| | | | | | | | | | probabilidade > 1.675
| | | | | | | | | | | probabilidade <= 1.761
| | | | | | | | | | | | LCSR <= 0.3: no (18.0)
| | | | | | | | | | | | LCSR > 0.3
| | | | | | | | | | | | | LCSR <= 0.455: yes (12.0)
| | | | | | | | | | | | | LCSR > 0.455: no (6.0)
| | | | | | | | | | | probabilidade > 1.761
| | | | | | | | | | | | probabilidade <= 2.995
| | | | | | | | | | | | | probabilidade <= 1.998
| | | | | | | | | | | | | | probabilidade <= 1.997: yes (30.0)
| | | | | | | | | | | | | | probabilidade > 1.997: no (6.0)
| | | | | | | | | | | | | probabilidade > 1.998: yes (72.0)
| | | | | | | | | | | | probabilidade > 2.995
| | | | | | | | | | | | | probabilidade <= 4.698
| | | | | | | | | | | | | | probabilidade <= 3.199: no (12.0)
| | | | | | | | | | | | | | probabilidade > 3.199
| | | | | | | | | | | | | | | probabilidade <= 4.024
| | | | | | | | | | | | | | | | probabilidade <= 3.578
| | | | | | | | | | | | | | | | | probabilidade <= 3.299: yes (6.0)
| | | | | | | | | | | | | | | | | probabilidade > 3.299
| | | | | | | | | | | | | | | | | | probabilidade <= 3.409: no (12.0)
| | | | | | | | | | | | | | | | | | probabilidade > 3.409
| | | | | | | | | | | | | | | | | | | LCSR <= 0.222: no (6.0)
| | | | | | | | | | | | | | | | | | | LCSR > 0.222: yes (6.0)
| | | | | | | | | | | | | | | | probabilidade > 3.578
| | | | | | | | | | | | | | | | | LCSR <= 0.3: yes (18.0)
| | | | | | | | | | | | | | | | | LCSR > 0.3
| | | | | | | | | | | | | | | | | | LCSR <= 0.455: no (6.0)
| | | | | | | | | | | | | | | | | | LCSR > 0.455: yes (6.0)
| | | | | | | | | | | | | | | probabilidade > 4.024
| | | | | | | | | | | | | | | | probabilidade <= 4.373: no (18.0)
| | | | | | | | | | | | | | | | probabilidade > 4.373
| | | | | | | | | | | | | | | | | LCSR <= 0.286: yes (6.0)
| | | | | | | | | | | | | | | | | LCSR > 0.286: no (6.0)
| | | | | | | | | | | | | probabilidade > 4.698
| | | | | | | | | | | | | | probabilidade <= 4.954: yes (36.0)
| | | | | | | | | | | | | | probabilidade > 4.954
| | | | | | | | | | | | | | | probabilidade <= 5.051: no (12.0)
| | | | | | | | | | | | | | | probabilidade > 5.051
| | | | | | | | | | | | | | | | probabilidade <= 5.86: yes (24.0)
| | | | | | | | | | | | | | | | probabilidade > 5.86: no (6.0)
| | | | | | | | LCSR > 0.545
| | | | | | | | | LCSR <= 0.667: no (30.0)
| | | | | | | | | LCSR > 0.667: yes (6.0)
| probabilidade > 6.198: yes (114.0/6.0)
```

```
probabilidade > 9.475
|  probabilidade <= 14.049
|  |  probabilidade <= 12.94
|  |  |  LCSR <= 0.182
|  |  |  |  probabilidade <= 10.292: no (38.0)
|  |  |  |  probabilidade > 10.292
|  |  |  |  |  probabilidade <= 10.949
|  |  |  |  |  |  LCSR <= 0.077: no (16.0)
|  |  |  |  |  |  LCSR > 0.077
|  |  |  |  |  |  |  probabilidade <= 10.925
|  |  |  |  |  |  |  |  probabilidade <= 10.885
|  |  |  |  |  |  |  |  |  probabilidade <= 10.727
|  |  |  |  |  |  |  |  |  |  LCSR <= 0.083: no (6.0)
|  |  |  |  |  |  |  |  |  |  LCSR > 0.083
|  |  |  |  |  |  |  |  |  |  |  LCSR <= 0.143
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade <= 10.455: yes (17.0)
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade > 10.455
|  |  |  |  |  |  |  |  |  |  |  |  |  LCSR <= 0.1: yes (5.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  LCSR > 0.1: no (15.0)
|  |  |  |  |  |  |  |  |  |  |  LCSR > 0.143
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade <= 10.42: no (10.0)
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade > 10.42: yes (5.0)
|  |  |  |  |  |  |  |  |  |  probabilidade > 10.727: yes (16.0)
|  |  |  |  |  |  |  |  |  probabilidade > 10.885: no (12.0)
|  |  |  |  |  |  |  |  probabilidade > 10.925: yes (16.0)
|  |  |  |  |  probabilidade > 10.949
|  |  |  |  |  |  probabilidade <= 12.285
|  |  |  |  |  |  |  probabilidade <= 11.241: no (36.0)
|  |  |  |  |  |  |  probabilidade > 11.241
|  |  |  |  |  |  |  |  probabilidade <= 11.574
|  |  |  |  |  |  |  |  |  probabilidade <= 11.474
|  |  |  |  |  |  |  |  |  |  probabilidade <= 11.253: yes (5.0)
|  |  |  |  |  |  |  |  |  |  probabilidade > 11.253: no (26.0)
|  |  |  |  |  |  |  |  |  probabilidade > 11.474: yes (10.0)
|  |  |  |  |  |  |  |  probabilidade > 11.574
|  |  |  |  |  |  |  |  |  probabilidade <= 12.089: no (38.0)
|  |  |  |  |  |  |  |  |  probabilidade > 12.089
|  |  |  |  |  |  |  |  |  |  probabilidade <= 12.09: yes (5.0)
|  |  |  |  |  |  |  |  |  |  probabilidade > 12.09: no (27.0)
|  |  |  |  |  |  probabilidade > 12.285
|  |  |  |  |  |  |  probabilidade <= 12.838
|  |  |  |  |  |  |  |  probabilidade <= 12.605
|  |  |  |  |  |  |  |  |  LCSR <= 0.125
|  |  |  |  |  |  |  |  |  |  probabilidade <= 12.381: no (5.0)
|  |  |  |  |  |  |  |  |  |  probabilidade > 12.381
|  |  |  |  |  |  |  |  |  |  |  probabilidade <= 12.479: yes (6.0)
|  |  |  |  |  |  |  |  |  |  |  probabilidade > 12.479
|  |  |  |  |  |  |  |  |  |  |  |  LCSR <= 0.1: no (6.0)
|  |  |  |  |  |  |  |  |  |  |  |  LCSR > 0.1: yes (5.0)
|  |  |  |  |  |  |  |  |  LCSR > 0.125: yes (5.0)
|  |  |  |  |  |  |  |  probabilidade > 12.605: no (10.0)
|  |  |  |  |  |  |  probabilidade > 12.838: yes (10.0)
|  |  |  LCSR > 0.182
|  |  |  |  probabilidade <= 11.275
|  |  |  |  |  probabilidade <= 11.099
|  |  |  |  |  |  probabilidade <= 10.925
|  |  |  |  |  |  |  probabilidade <= 10.694
|  |  |  |  |  |  |  |  probabilidade <= 10.589
|  |  |  |  |  |  |  |  |  probabilidade <= 10.041: no (5.0)
|  |  |  |  |  |  |  |  |  probabilidade > 10.041
|  |  |  |  |  |  |  |  |  |  LCSR <= 0.222: yes (12.0)
|  |  |  |  |  |  |  |  |  |  LCSR > 0.222
|  |  |  |  |  |  |  |  |  |  |  probabilidade <= 10.57
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade <= 10.136: yes (11.0)
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade > 10.136: no (11.0)
```

```
| | | | | | | | | | | probabilidade > 10.57: yes (12.0)
| | | | | | | | | probabilidade > 10.589: no (16.0)
| | | | | | | | probabilidade > 10.694: yes (15.0)
| | | | | | | probabilidade > 10.925: no (11.0)
| | | | | | probabilidade > 11.099: yes (17.0)
| | | | | probabilidade > 11.275
| | | | | | LCSR <= 0.333
| | | | | | | probabilidade <= 12.09
| | | | | | | | LCSR <= 0.3: no (60.0)
| | | | | | | | LCSR > 0.3
| | | | | | | | | probabilidade <= 11.637: no (6.0)
| | | | | | | | | probabilidade > 11.637
| | | | | | | | | | probabilidade <= 11.765: yes (6.0)
| | | | | | | | | | probabilidade > 11.765: no (5.0)
| | | | | | | probabilidade > 12.09
| | | | | | | | LCSR <= 0.222
| | | | | | | | | probabilidade <= 12.517: yes (16.0)
| | | | | | | | | probabilidade > 12.517
| | | | | | | | | | probabilidade <= 12.539: no (5.0)
| | | | | | | | | | probabilidade > 12.539: yes (10.0)
| | | | | | | | LCSR > 0.222
| | | | | | | | | probabilidade <= 12.285: yes (5.0)
| | | | | | | | | probabilidade > 12.285: no (37.0)
| | | | | | LCSR > 0.333
| | | | | | | probabilidade <= 12.267: yes (20.0)
| | | | | | | probabilidade > 12.267
| | | | | | | | LCSR <= 0.385: yes (5.0)
| | | | | | | | LCSR > 0.385: no (10.0)
| | | probabilidade > 12.94
| | | | probabilidade <= 13.244: no (57.0)
| | | | probabilidade > 13.244
| | | | | LCSR <= 0.182
| | | | | | probabilidade <= 13.246: yes (5.0)
| | | | | | probabilidade > 13.246
| | | | | | | probabilidade <= 13.817: no (78.0)
| | | | | | | probabilidade > 13.817
| | | | | | | | LCSR <= 0.111: yes (5.0)
| | | | | | | | LCSR > 0.111: no (25.0)
| | | | | LCSR > 0.182
| | | | | | LCSR <= 0.2
| | | | | | | probabilidade <= 13.428: no (5.0)
| | | | | | | probabilidade > 13.428
| | | | | | | | probabilidade <= 13.576: yes (5.0)
| | | | | | | | probabilidade > 13.576
| | | | | | | | | probabilidade <= 13.684: no (5.0)
| | | | | | | | | probabilidade > 13.684: yes (5.0)
| | | | | | LCSR > 0.2
| | | | | | | probabilidade <= 13.602
| | | | | | | | LCSR <= 0.222: yes (6.0)
| | | | | | | | LCSR > 0.222
| | | | | | | | | probabilidade <= 13.294: no (5.0)
| | | | | | | | | probabilidade > 13.294
| | | | | | | | | | probabilidade <= 13.354: yes (5.0)
| | | | | | | | | | probabilidade > 13.354
| | | | | | | | | | | probabilidade <= 13.38: no (5.0)
| | | | | | | | | | | probabilidade > 13.38
| | | | | | | | | | | | probabilidade <= 13.428: yes (5.0)
| | | | | | | | | | | | probabilidade > 13.428: no (11.0)
| | | | | | probabilidade > 13.602: no (35.0)
| probabilidade > 14.049
| | probabilidade <= 14.201: yes (27.0)
| | probabilidade > 14.201
| | | probabilidade <= 14.57: no (36.0)
| | | probabilidade > 14.57
| | | | LCSR <= 0.571
```

```
| | | | | probabilidade <= 64.946
| | | | | | LCSR <= 0.2
| | | | | | | LCSR <= 0.182: no (1076.0/429.0)
| | | | | | | LCSR > 0.182
| | | | | | | | probabilidade <= 41.307
| | | | | | | | | probabilidade <= 19.877
| | | | | | | | | | probabilidade <= 16.332: no (17.0)
| | | | | | | | | | probabilidade > 16.332
| | | | | | | | | | | probabilidade <= 17.607: yes (11.0)
| | | | | | | | | | | probabilidade > 17.607
| | | | | | | | | | | | probabilidade <= 18.375: no (16.0)
| | | | | | | | | | | | probabilidade > 18.375
| | | | | | | | | | | | | probabilidade <= 18.824: yes (5.0)
| | | | | | | | | | | | | probabilidade > 18.824
| | | | | | | | | | | | | | probabilidade <= 19.466: no (5.0)
| | | | | | | | | | | | | | probabilidade > 19.466
| | | | | | | | | | | | | | | probabilidade <= 19.577: yes (5.0)
| | | | | | | | | | | | | | | probabilidade > 19.577
| | | | | | | | | | | | | | | | probabilidade <= 19.712: no (5.0)
| | | | | | | | | | | | | | | | probabilidade > 19.712: yes (5.0)
| | | | | | | | | probabilidade > 19.877
| | | | | | | | | | probabilidade <= 24.397: no (40.0)
| | | | | | | | | | probabilidade > 24.397
| | | | | | | | | | | probabilidade <= 26.357: yes (10.0)
| | | | | | | | | | | probabilidade > 26.357
| | | | | | | | | | | | probabilidade <= 34.636: no (35.0)
| | | | | | | | | | | | probabilidade > 34.636
| | | | | | | | | | | | | probabilidade <= 37.969: yes (6.0)
| | | | | | | | | | | | | probabilidade > 37.969: no (11.0)
| | | | | | | | probabilidade > 41.307
| | | | | | | | | probabilidade <= 42.295: yes (6.0)
| | | | | | | | | probabilidade > 42.295
| | | | | | | | | | probabilidade <= 44.113: no (6.0)
| | | | | | | | | | probabilidade > 44.113
| | | | | | | | | | | probabilidade <= 47.599: yes (10.0)
| | | | | | | | | | | probabilidade > 47.599
| | | | | | | | | | | | probabilidade <= 52.174: no (5.0)
| | | | | | | | | | | | probabilidade > 52.174: yes (6.0)
| | | | | | LCSR > 0.2
| | | | | | | LCSR <= 0.25
| | | | | | | | probabilidade <= 44.788
| | | | | | | | | probabilidade <= 37.487
| | | | | | | | | | LCSR <= 0.222
| | | | | | | | | | | probabilidade <= 31.269
| | | | | | | | | | | | probabilidade <= 21.429
| | | | | | | | | | | | | probabilidade <= 18.824
| | | | | | | | | | | | | | probabilidade <= 15.973: no (11.0)
| | | | | | | | | | | | | | probabilidade > 15.973
| | | | | | | | | | | | | | | probabilidade <= 17.442: yes (6.0)
| | | | | | | | | | | | | | | probabilidade > 17.442: no (6.0)
| | | | | | | | | | | | | | probabilidade > 18.824: yes (16.0)
| | | | | | | | | | | | | probabilidade > 21.429: no (22.0)
| | | | | | | | | | | | probabilidade > 31.269: yes (15.0)
| | | | | | | | | | | LCSR > 0.222
| | | | | | | | | | | | probabilidade <= 35.926
| | | | | | | | | | | | | probabilidade <= 25.127
| | | | | | | | | | | | | | probabilidade <= 24.742
| | | | | | | | | | | | | | | probabilidade <= 24.603
| | | | | | | | | | | | | | | | probabilidade <= 22.787
| | | | | | | | | | | | | | | | | probabilidade <= 21.636
| | | | | | | | | | | | | | | | | | probabilidade <= 19.877
| | | | | | | | | | | | | | | | | | | probabilidade <= 19.256
| | | | | | | | | | | | | | | | | | | | probabilidade <= 16.828
| | | | | | | | | | | | | | | | | | | | | probabilidade <= 16.056: yes (6.0)
| | | | | | | | | | | | | | | | | | | | | probabilidade > 16.056: no (6.0)
```

```
| | | | | | | | | | | | | | | | | | | | probabilidade > 16.828: yes (16.0)
| | | | | | | | | | | | | | | | | | | | probabilidade > 19.256: no (5.0)
| | | | | | | | | | | | | | | | | | | probabilidade > 19.877: yes (10.0)
| | | | | | | | | | | | | | | | | | | probabilidade > 21.636: no (5.0)
| | | | | | | | | | | | | | | | | | | probabilidade > 22.787: yes (15.0)
| | | | | | | | | | | | | | | | | | | probabilidade > 24.603: no (5.0)
| | | | | | | | | | | | | | | | | | | probabilidade > 24.742: yes (17.0)
| | | | | | | | | | | | | | | | | | | probabilidade > 25.127
| | | | | | | | | | | | | | | | | | | probabilidade <= 26.743: no (5.0)
| | | | | | | | | | | | | | | | | | | probabilidade > 26.743
| | | | | | | | | | | | | | | | | | | | probabilidade <= 35.013
| | | | | | | | | | | | | | | | | | | | probabilidade <= 32.712
| | | | | | | | | | | | | | | | | | | | | probabilidade <= 31.048
| | | | | | | | | | | | | | | | | | | | | probabilidade <= 29.677
| | | | | | | | | | | | | | | | | | | | | | probabilidade <= 28.571: yes (5.0)
| | | | | | | | | | | | | | | | | | | | | | probabilidade > 28.571
| | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 28.849: no (6.0)
| | | | | | | | | | | | | | | | | | | | | | | probabilidade > 28.849
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 29.159: yes (5.0)
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 29.159: no (6.0)
| | | | | | | | | | | | | | | | | | | | | probabilidade > 29.677: yes (10.0)
| | | | | | | | | | | | | | | | | | | | | probabilidade > 31.048: no (6.0)
| | | | | | | | | | | | | | | | | | | | | probabilidade > 32.712: yes (6.0)
| | | | | | | | | | | | | | | | | | | | | probabilidade > 35.013: no (5.0)
| | | | | | | | | | | | | | | | | | | | | probabilidade > 35.926: yes (10.0)
| | | | | | | | | | | | | | | | | | | | probabilidade > 37.487
| | | | | | | | | | | | | | | | | | | | | LCSR <= 0.222
| | | | | | | | | | | | | | | | | | | | | | probabilidade <= 42.295
| | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 39.189: yes (6.0)
| | | | | | | | | | | | | | | | | | | | | | | probabilidade > 39.189
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 40: no (5.0)
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 40: yes (6.0)
| | | | | | | | | | | | | | | | | | | | | | | probabilidade > 42.295: no (6.0)
| | | | | | | | | | | | | | | | | | | | | | LCSR > 0.222: no (31.0)
| | | | | | | | | | | | | | | | | | | | probabilidade > 44.788
| | | | | | | | | | | | | | | | | | | | | probabilidade <= 52.174: yes (49.0)
| | | | | | | | | | | | | | | | | | | | | probabilidade > 52.174
| | | | | | | | | | | | | | | | | | | | | | LCSR <= 0.222: no (11.0)
| | | | | | | | | | | | | | | | | | | | | | LCSR > 0.222
| | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 60.694
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 54.092
| | | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 53.107: no (6.0)
| | | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 53.107: yes (5.0)
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 54.092: no (10.0)
| | | | | | | | | | | | | | | | | | | | | | | probabilidade > 60.694: yes (11.0)
| | | | | | | | | | | | | | | | | | | | | LCSR > 0.25
| | | | | | | | | | | | | | | | | | | | | probabilidade <= 15.805
| | | | | | | | | | | | | | | | | | | | | | probabilidade <= 15.187
| | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 15.054: yes (10.0)
| | | | | | | | | | | | | | | | | | | | | | | probabilidade > 15.054: no (5.0)
| | | | | | | | | | | | | | | | | | | | | | probabilidade > 15.187: yes (17.0)
| | | | | | | | | | | | | | | | | | | | | probabilidade > 15.805
| | | | | | | | | | | | | | | | | | | | | | LCSR <= 0.333
| | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 33.128
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 28.29: no (142.0/44.0)
| | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 28.29: yes (10.0)
| | | | | | | | | | | | | | | | | | | | | | | probabilidade > 33.128
| | | | | | | | | | | | | | | | | | | | | | | LCSR <= 0.3
| | | | | | | | | | | | | | | | | | | | | | | | LCSR <= 0.286
| | | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 39.922: no (22.0)
| | | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 39.922
| | | | | | | | | | | | | | | | | | | | | | | | | | probabilidade <= 41.964: yes (5.0)
| | | | | | | | | | | | | | | | | | | | | | | | | | probabilidade > 41.964: no (22.0)
| | | | | | | | | | | | | | | | | | | | | | | | LCSR > 0.286: yes (10.0)
| | | | | | | | | | | | | | | | | | | | | | | LCSR > 0.3: no (46.0)
```

```
| | | | | | | | | LCSR > 0.333
| | | | | | | | | | probabilidade <= 21.798
| | | | | | | | | | | probabilidade <= 17.442
| | | | | | | | | | | | probabilidade <= 16.814
| | | | | | | | | | | | | LCSR <= 0.417: no (16.0)
| | | | | | | | | | | | | LCSR > 0.417: yes (5.0)
| | | | | | | | | | | | probabilidade > 16.814: yes (11.0)
| | | | | | | | | | | probabilidade > 17.442: no (47.0)
| | | | | | | | | | probabilidade > 21.798
| | | | | | | | | | | probabilidade <= 22.444: yes (16.0)
| | | | | | | | | | | probabilidade > 22.444
| | | | | | | | | | | | probabilidade <= 39.432
| | | | | | | | | | | | | LCSR <= 0.545
| | | | | | | | | | | | | | LCSR <= 0.5
| | | | | | | | | | | | | | | probabilidade <= 27.778
| | | | | | | | | | | | | | | | probabilidade <= 23.558
| | | | | | | | | | | | | | | | | probabilidade <= 22.622: no (5.0)
| | | | | | | | | | | | | | | | | probabilidade > 22.622: yes (10.0)
| | | | | | | | | | | | | | | | probabilidade > 23.558: no (21.0)
| | | | | | | | | | | | | | | probabilidade > 27.778
| | | | | | | | | | | | | | | | probabilidade <= 35.926: yes (10.0)
| | | | | | | | | | | | | | | | probabilidade > 35.926
| | | | | | | | | | | | | | | | | LCSR <= 0.4
| | | | | | | | | | | | | | | | | | probabilidade <= 37.969: yes (6.0)
| | | | | | | | | | | | | | | | | | probabilidade > 37.969: no (5.0)
| | | | | | | | | | | | | | | | | LCSR > 0.4: no (5.0)
| | | | | | | | | | | | | | LCSR > 0.5: yes (5.0)
| | | | | | | | | | | | | LCSR > 0.545: no (5.0)
| | | | | | | | | | | | probabilidade > 39.432
| | | | | | | | | | | | | probabilidade <= 61.782: yes (21.0)
| | | | | | | | | | | | | probabilidade > 61.782: no (6.0)
| | | | | probabilidade > 64.946
| | | | | | LCSR <= 0.214: yes (32.0)
| | | | | | LCSR > 0.214
| | | | | | | LCSR <= 0.286: no (26.0)
| | | | | | | LCSR > 0.286
| | | | | | | | LCSR <= 0.444: yes (21.0)
| | | | | | | | LCSR > 0.444: no (5.0)
| | | | LCSR > 0.571
| | | | | LCSR <= 0.625: no (31.0)
| | | | | LCSR > 0.625
| | | | | | LCSR <= 0.636: yes (10.0)
| | | | | | LCSR > 0.636
| | | | | | | probabilidade <= 23.558: yes (5.0)
| | | | | | | probabilidade > 23.558
| | | | | | | | LCSR <= 0.75: no (22.0)
| | | | | | | | LCSR > 0.75
| | | | | | | | | LCSR <= 0.8: yes (5.0)
| | | | | | | | | LCSR > 0.8: no (12.0)
```

Árvore gerada durante treinamento do Promotor-2:

```
Test mode:10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

probabilidade <= 9.475
|  probabilidade <= 6.198
|  |  LCSR <= 0.091
|  |  |  LCSR <= 0: no (11.0)
|  |  |  LCSR > 0
|  |  |  |  LCSR <= 0.077
```

```
| | | | | probabilidade <= 0.871
| | | | | | probabilidade <= 0.84
| | | | | | | probabilidade <= 0.82: yes (5.0)
| | | | | | | probabilidade > 0.82: no (5.0)
| | | | | | probabilidade > 0.84: yes (6.0)
| | | | | probabilidade > 0.871: no (6.0)
| | | | LCSR > 0.077
| | | | | LCSR <= 0.083: no (16.0)
| | | | | LCSR > 0.083
| | | | | | probabilidade <= 0.669: yes (6.0)
| | | | | | probabilidade > 0.669
| | | | | | | probabilidade <= 0.735: no (15.0)
| | | | | | | probabilidade > 0.735
| | | | | | | | probabilidade <= 0.76: yes (5.0)
| | | | | | | | probabilidade > 0.76
| | | | | | | | | probabilidade <= 2.145: no (10.0)
| | | | | | | | | probabilidade > 2.145
| | | | | | | | | | probabilidade <= 3.704: yes (5.0)
| | | | | | | | | | probabilidade > 3.704: no (6.0)
| | LCSR > 0.091
| | | probabilidade <= 0.56: yes (29.0)
| | | probabilidade > 0.56
| | | | LCSR <= 0.1
| | | | | probabilidade <= 0.852: no (5.0)
| | | | | probabilidade > 0.852: yes (52.0)
| | | | LCSR > 0.1
| | | | | LCSR <= 0.154
| | | | | | probabilidade <= 1.338
| | | | | | | LCSR <= 0.125: no (30.0)
| | | | | | | LCSR > 0.125
| | | | | | | | probabilidade <= 1.141: yes (17.0)
| | | | | | | | probabilidade > 1.141: no (10.0)
| | | | | | probabilidade > 1.338
| | | | | | | probabilidade <= 1.784
| | | | | | | | probabilidade <= 1.612
| | | | | | | | | probabilidade <= 1.563
| | | | | | | | | | probabilidade <= 1.507
| | | | | | | | | | | probabilidade <= 1.409: yes (5.0)
| | | | | | | | | | | probabilidade > 1.409: no (5.0)
| | | | | | | | | | probabilidade > 1.507: yes (15.0)
| | | | | | | | | probabilidade > 1.563: no (6.0)
| | | | | | | | probabilidade > 1.612: yes (18.0)
| | | | | | | probabilidade > 1.784
| | | | | | | | probabilidade <= 2.461: no (15.0)
| | | | | | | | probabilidade > 2.461
| | | | | | | | | probabilidade <= 5.16
| | | | | | | | | | LCSR <= 0.111: yes (10.0)
| | | | | | | | | | LCSR > 0.111
| | | | | | | | | | | probabilidade <= 3.741
| | | | | | | | | | | | LCSR <= 0.125: yes (10.0)
| | | | | | | | | | | | LCSR > 0.125
| | | | | | | | | | | | | probabilidade <= 3.299: no (6.0)
| | | | | | | | | | | | | probabilidade > 3.299: yes (5.0)
| | | | | | | | | | probabilidade > 3.741
| | | | | | | | | | | probabilidade <= 4.846: no (22.0)
| | | | | | | | | | | probabilidade > 4.846: yes (5.0)
| | | | | | | | | probabilidade > 5.16: no (6.0)
| | | | | LCSR > 0.154
| | | | | | probabilidade <= 0.968
| | | | | | | probabilidade <= 0.809
| | | | | | | | probabilidade <= 0.793
| | | | | | | | | LCSR <= 0.267
| | | | | | | | | | probabilidade <= 0.565: no (5.0)
| | | | | | | | | | probabilidade > 0.565
| | | | | | | | | | | LCSR <= 0.2
```

```
| | | | | | | | | | | | probabilidade <= 0.672: no (10.0)
| | | | | | | | | | | | probabilidade > 0.672: yes (5.0)
| | | | | | | | | | | LCSR > 0.2: yes (22.0)
| | | | | | | | | | LCSR > 0.267: yes (26.0)
| | | | | | | | | probabilidade > 0.793: no (5.0)
| | | | | | | | probabilidade > 0.809: yes (53.0)
| | | | | | | probabilidade > 0.968
| | | | | | | | LCSR <= 0.545
| | | | | | | | | probabilidade <= 1.398
| | | | | | | | | | LCSR <= 0.417
| | | | | | | | | | | LCSR <= 0.182
| | | | | | | | | | | | probabilidade <= 0.998: no (5.0)
| | | | | | | | | | | | probabilidade > 0.998: yes (16.0)
| | | | | | | | | | | LCSR > 0.182
| | | | | | | | | | | | probabilidade <= 1.072
| | | | | | | | | | | | | probabilidade <= 1.025: no (15.0)
| | | | | | | | | | | | | probabilidade > 1.025: yes (11.0)
| | | | | | | | | | | | probabilidade > 1.072: no (36.0)
| | | | | | | | | | LCSR > 0.417: yes (22.0)
| | | | | | | | | probabilidade > 1.398
| | | | | | | | | | probabilidade <= 1.675: yes (44.0)
| | | | | | | | | | probabilidade > 1.675
| | | | | | | | | | | probabilidade <= 1.761
| | | | | | | | | | | | LCSR <= 0.3: no (17.0)
| | | | | | | | | | | | LCSR > 0.3
| | | | | | | | | | | | | LCSR <= 0.455: yes (11.0)
| | | | | | | | | | | | | LCSR > 0.455: no (6.0)
| | | | | | | | | | | probabilidade > 1.761
| | | | | | | | | | | | probabilidade <= 2.995
| | | | | | | | | | | | | probabilidade <= 1.998
| | | | | | | | | | | | | | probabilidade <= 1.997: yes (28.0)
| | | | | | | | | | | | | | probabilidade > 1.997: no (5.0)
| | | | | | | | | | | | | probabilidade > 1.998: yes (69.0)
| | | | | | | | | | | | probabilidade > 2.995
| | | | | | | | | | | | | probabilidade <= 3.199: no (12.0)
| | | | | | | | | | | | | probabilidade > 3.199
| | | | | | | | | | | | | | LCSR <= 0.3
| | | | | | | | | | | | | | | probabilidade <= 4.373
| | | | | | | | | | | | | | | | probabilidade <= 4.024
| | | | | | | | | | | | | | | | | probabilidade <= 3.578
| | | | | | | | | | | | | | | | | | LCSR <= 0.222: no (5.0)
| | | | | | | | | | | | | | | | | | LCSR > 0.222
| | | | | | | | | | | | | | | | | | | LCSR <= 0.273
| | | | | | | | | | | | | | | | | | | | probabilidade <= 3.409: no (5.0)
| | | | | | | | | | | | | | | | | | | | probabilidade > 3.409: yes (6.0)
| | | | | | | | | | | | | | | | | | | LCSR > 0.273: yes (6.0)
| | | | | | | | | | | | | | | | | | probabilidade > 3.578: yes (18.0)
| | | | | | | | | | | | | | | | | probabilidade > 4.024: no (16.0)
| | | | | | | | | | | | | | | probabilidade > 4.373
| | | | | | | | | | | | | | | | probabilidade <= 5.86
| | | | | | | | | | | | | | | | | probabilidade <= 4.954: yes (40.0)
| | | | | | | | | | | | | | | | | probabilidade > 4.954
| | | | | | | | | | | | | | | | | | probabilidade <= 5.051: no (5.0)
| | | | | | | | | | | | | | | | | | probabilidade > 5.051: yes (17.0)
| | | | | | | | | | | | | | | | probabilidade > 5.86: no (5.0)
| | | | | | | | | | | | | | LCSR > 0.3
| | | | | | | | | | | | | | | LCSR <= 0.455
| | | | | | | | | | | | | | | | probabilidade <= 5.16: no (22.0)
| | | | | | | | | | | | | | | | probabilidade > 5.16: yes (5.0)
| | | | | | | | | | | | | | | LCSR > 0.455: yes (5.0)
| | | | | | | | LCSR > 0.545
| | | | | | | | | LCSR <= 0.667: no (26.0)
| | | | | | | | | LCSR > 0.667: yes (5.0)
| probabilidade > 6.198: yes (103.0/5.0)
probabilidade > 9.475
```

```
|  probabilidade <= 14.049
|  |  probabilidade <= 12.94
|  |  |  LCSR <= 0.182
|  |  |  |  probabilidade <= 10.292: no (37.0)
|  |  |  |  probabilidade > 10.292
|  |  |  |  |  probabilidade <= 10.949
|  |  |  |  |  |  LCSR <= 0.077: no (16.0)
|  |  |  |  |  |  LCSR > 0.077
|  |  |  |  |  |  |  probabilidade <= 10.925
|  |  |  |  |  |  |  |  probabilidade <= 10.885
|  |  |  |  |  |  |  |  |  probabilidade <= 10.727
|  |  |  |  |  |  |  |  |  |  LCSR <= 0.083: no (6.0)
|  |  |  |  |  |  |  |  |  |  LCSR > 0.083
|  |  |  |  |  |  |  |  |  |  |  LCSR <= 0.143
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade <= 10.455: yes (17.0)
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade > 10.455
|  |  |  |  |  |  |  |  |  |  |  |  |  LCSR <= 0.1: yes (5.0)
|  |  |  |  |  |  |  |  |  |  |  |  |  LCSR > 0.1: no (15.0)
|  |  |  |  |  |  |  |  |  |  |  LCSR > 0.143
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade <= 10.42: no (10.0)
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade > 10.42: yes (5.0)
|  |  |  |  |  |  |  |  |  probabilidade > 10.727: yes (15.0)
|  |  |  |  |  |  |  |  probabilidade > 10.885: no (12.0)
|  |  |  |  |  |  |  probabilidade > 10.925: yes (15.0)
|  |  |  |  |  probabilidade > 10.949
|  |  |  |  |  |  probabilidade <= 12.838
|  |  |  |  |  |  |  probabilidade <= 11.241: no (35.0)
|  |  |  |  |  |  |  probabilidade > 11.241
|  |  |  |  |  |  |  |  probabilidade <= 12.285
|  |  |  |  |  |  |  |  |  probabilidade <= 11.574
|  |  |  |  |  |  |  |  |  |  probabilidade <= 11.474
|  |  |  |  |  |  |  |  |  |  |  probabilidade <= 11.253: yes (5.0)
|  |  |  |  |  |  |  |  |  |  |  probabilidade > 11.253: no (26.0)
|  |  |  |  |  |  |  |  |  |  probabilidade > 11.474: yes (10.0)
|  |  |  |  |  |  |  |  |  probabilidade > 11.574
|  |  |  |  |  |  |  |  |  |  probabilidade <= 12.089: no (36.0)
|  |  |  |  |  |  |  |  |  |  probabilidade > 12.089
|  |  |  |  |  |  |  |  |  |  |  probabilidade <= 12.09: yes (5.0)
|  |  |  |  |  |  |  |  |  |  |  probabilidade > 12.09: no (27.0)
|  |  |  |  |  |  |  |  probabilidade > 12.285
|  |  |  |  |  |  |  |  probabilidade <= 12.605
|  |  |  |  |  |  |  |  |  LCSR <= 0.125
|  |  |  |  |  |  |  |  |  |  probabilidade <= 12.381: no (5.0)
|  |  |  |  |  |  |  |  |  |  probabilidade > 12.381
|  |  |  |  |  |  |  |  |  |  |  probabilidade <= 12.479: yes (6.0)
|  |  |  |  |  |  |  |  |  |  |  probabilidade > 12.479
|  |  |  |  |  |  |  |  |  |  |  |  LCSR <= 0.1: no (6.0)
|  |  |  |  |  |  |  |  |  |  |  |  LCSR > 0.1: yes (5.0)
|  |  |  |  |  |  |  |  |  LCSR > 0.125: yes (5.0)
|  |  |  |  |  |  |  |  probabilidade > 12.605: no (10.0)
|  |  |  |  |  |  probabilidade > 12.838: yes (10.0)
|  |  |  LCSR > 0.182
|  |  |  |  probabilidade <= 11.275
|  |  |  |  |  probabilidade <= 11.099
|  |  |  |  |  |  probabilidade <= 10.925
|  |  |  |  |  |  |  probabilidade <= 10.694
|  |  |  |  |  |  |  |  probabilidade <= 10.589
|  |  |  |  |  |  |  |  |  probabilidade <= 10.041: no (5.0)
|  |  |  |  |  |  |  |  |  probabilidade > 10.041
|  |  |  |  |  |  |  |  |  |  LCSR <= 0.222: yes (11.0)
|  |  |  |  |  |  |  |  |  |  LCSR > 0.222
|  |  |  |  |  |  |  |  |  |  |  probabilidade <= 10.57
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade <= 10.136: yes (10.0)
|  |  |  |  |  |  |  |  |  |  |  |  probabilidade > 10.136: no (10.0)
|  |  |  |  |  |  |  |  |  |  |  probabilidade > 10.57: yes (12.0)
```

```
| | | | | | | | probabilidade > 10.589: no (16.0)
| | | | | | | probabilidade > 10.694: yes (15.0)
| | | | | | probabilidade > 10.925: no (11.0)
| | | | | probabilidade > 11.099: yes (17.0)
| | | | probabilidade > 11.275
| | | | | LCSR <= 0.333
| | | | | | probabilidade <= 12.09
| | | | | | | LCSR <= 0.3: no (58.0)
| | | | | | | LCSR > 0.3
| | | | | | | | probabilidade <= 11.637: no (5.0)
| | | | | | | | probabilidade > 11.637
| | | | | | | | | probabilidade <= 11.765: yes (6.0)
| | | | | | | | | probabilidade > 11.765: no (5.0)
| | | | | | probabilidade > 12.09
| | | | | | | LCSR <= 0.222
| | | | | | | | probabilidade <= 12.517: yes (16.0)
| | | | | | | | probabilidade > 12.517
| | | | | | | | | probabilidade <= 12.539: no (5.0)
| | | | | | | | | probabilidade > 12.539: yes (10.0)
| | | | | | | LCSR > 0.222
| | | | | | | | probabilidade <= 12.285: yes (5.0)
| | | | | | | | probabilidade > 12.285: no (36.0)
| | | | | LCSR > 0.333
| | | | | | probabilidade <= 12.267: yes (20.0)
| | | | | | probabilidade > 12.267
| | | | | | | LCSR <= 0.385: yes (5.0)
| | | | | | | LCSR > 0.385: no (10.0)
| | probabilidade > 12.94
| | | probabilidade <= 13.244: no (57.0)
| | | probabilidade > 13.244
| | | | LCSR <= 0.182
| | | | | probabilidade <= 13.246: yes (5.0)
| | | | | probabilidade > 13.246
| | | | | | probabilidade <= 13.817: no (77.0)
| | | | | | probabilidade > 13.817
| | | | | | | LCSR <= 0.111: yes (5.0)
| | | | | | | LCSR > 0.111: no (25.0)
| | | | LCSR > 0.182
| | | | | LCSR <= 0.2
| | | | | | probabilidade <= 13.428: no (5.0)
| | | | | | probabilidade > 13.428
| | | | | | | probabilidade <= 13.576: yes (5.0)
| | | | | | | probabilidade > 13.576
| | | | | | | | probabilidade <= 13.684: no (5.0)
| | | | | | | | probabilidade > 13.684: yes (5.0)
| | | | | LCSR > 0.2
| | | | | | probabilidade <= 13.602
| | | | | | | LCSR <= 0.222: yes (6.0)
| | | | | | | LCSR > 0.222
| | | | | | | | probabilidade <= 13.294: no (5.0)
| | | | | | | | probabilidade > 13.294
| | | | | | | | | probabilidade <= 13.354: yes (5.0)
| | | | | | | | | probabilidade > 13.354
| | | | | | | | | | probabilidade <= 13.38: no (5.0)
| | | | | | | | | | probabilidade > 13.38
| | | | | | | | | | | probabilidade <= 13.428: yes (5.0)
| | | | | | | | | | | probabilidade > 13.428: no (11.0)
| | | | | probabilidade > 13.602: no (35.0)
| probabilidade > 14.049
| | probabilidade <= 14.201: yes (25.0)
| | probabilidade > 14.201
| | | probabilidade <= 14.57: no (35.0)
| | | probabilidade > 14.57
| | | | LCSR <= 0.571
| | | | | probabilidade <= 64.946
```

```
|   |   |   |   |   |   LCSR <= 0.2: no (1256.0/479.0)
|   |   |   |   |   |   LCSR > 0.2
|   |   |   |   |   |   |   LCSR <= 0.25
|   |   |   |   |   |   |   |   probabilidade <= 44.788
|   |   |   |   |   |   |   |   |   probabilidade <= 37.487
|   |   |   |   |   |   |   |   |   |   probabilidade <= 31.667
|   |   |   |   |   |   |   |   |   |   |   probabilidade <= 30.859
|   |   |   |   |   |   |   |   |   |   |   |   LCSR <= 0.222
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 15.973: no (11.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 15.973
|   |   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 21.429
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 18.824
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 17.442: yes (6.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 17.442: no (6.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 18.824: yes (16.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 21.429: no (10.0)
|   |   |   |   |   |   |   |   |   |   |   |   LCSR > 0.222
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 29.677
|   |   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 29.159: yes (104.0/32.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 29.159: no (6.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 29.677: yes (10.0)
|   |   |   |   |   |   |   |   |   |   |   probabilidade > 30.859: no (17.0)
|   |   |   |   |   |   |   |   |   |   probabilidade > 31.667
|   |   |   |   |   |   |   |   |   |   |   probabilidade <= 35.013: yes (21.0)
|   |   |   |   |   |   |   |   |   |   |   probabilidade > 35.013
|   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 35.926: no (5.0)
|   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 35.926: yes (10.0)
|   |   |   |   |   |   |   |   |   probabilidade > 37.487
|   |   |   |   |   |   |   |   |   |   LCSR <= 0.222
|   |   |   |   |   |   |   |   |   |   |   probabilidade <= 42.295
|   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 40
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 39.189: yes (5.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 39.189: no (5.0)
|   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 40: yes (6.0)
|   |   |   |   |   |   |   |   |   |   |   probabilidade > 42.295: no (5.0)
|   |   |   |   |   |   |   |   |   |   LCSR > 0.222: no (31.0)
|   |   |   |   |   |   |   |   probabilidade > 44.788
|   |   |   |   |   |   |   |   |   probabilidade <= 52.174: yes (48.0)
|   |   |   |   |   |   |   |   |   probabilidade > 52.174
|   |   |   |   |   |   |   |   |   |   LCSR <= 0.222: no (10.0)
|   |   |   |   |   |   |   |   |   |   LCSR > 0.222
|   |   |   |   |   |   |   |   |   |   |   probabilidade <= 60.694
|   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 54.092
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 53.107: no (6.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 53.107: yes (5.0)
|   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 54.092: no (10.0)
|   |   |   |   |   |   |   |   |   |   |   probabilidade > 60.694: yes (11.0)
|   |   |   |   |   |   |   LCSR > 0.25
|   |   |   |   |   |   |   |   probabilidade <= 15.805
|   |   |   |   |   |   |   |   |   probabilidade <= 15.187
|   |   |   |   |   |   |   |   |   |   probabilidade <= 15.054: yes (10.0)
|   |   |   |   |   |   |   |   |   |   probabilidade > 15.054: no (5.0)
|   |   |   |   |   |   |   |   |   probabilidade > 15.187: yes (15.0)
|   |   |   |   |   |   |   |   probabilidade > 15.805
|   |   |   |   |   |   |   |   |   LCSR <= 0.333
|   |   |   |   |   |   |   |   |   |   probabilidade <= 41.63
|   |   |   |   |   |   |   |   |   |   |   probabilidade <= 39.417
|   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 33.128
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade <= 28.29: no (138.0/42.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 28.29: yes (10.0)
|   |   |   |   |   |   |   |   |   |   |   |   probabilidade > 33.128: no (36.0)
|   |   |   |   |   |   |   |   |   |   |   probabilidade > 39.417: yes (10.0)
|   |   |   |   |   |   |   |   |   |   probabilidade > 41.63
|   |   |   |   |   |   |   |   |   |   |   probabilidade <= 61.483: no (48.0)
|   |   |   |   |   |   |   |   |   |   |   probabilidade > 61.483
```

```
| | | | | | | | | | | | LCSR <= 0.286: no (5.0)
| | | | | | | | | | | | LCSR > 0.286: yes (5.0)
| | | | | | | | | LCSR > 0.333
| | | | | | | | | | probabilidade <= 21.798
| | | | | | | | | | | probabilidade <= 17.442
| | | | | | | | | | | | probabilidade <= 16.814
| | | | | | | | | | | | | LCSR <= 0.417: no (16.0)
| | | | | | | | | | | | | LCSR > 0.417: yes (5.0)
| | | | | | | | | | | | probabilidade > 16.814: yes (10.0)
| | | | | | | | | | | probabilidade > 17.442: no (46.0)
| | | | | | | | | | probabilidade > 21.798
| | | | | | | | | | | probabilidade <= 22.444: yes (16.0)
| | | | | | | | | | | probabilidade > 22.444
| | | | | | | | | | | | probabilidade <= 39.432
| | | | | | | | | | | | | LCSR <= 0.545
| | | | | | | | | | | | | | LCSR <= 0.5
| | | | | | | | | | | | | | | probabilidade <= 27.778
| | | | | | | | | | | | | | | | probabilidade <= 23.558
| | | | | | | | | | | | | | | | | probabilidade <= 22.622: no (5.0)
| | | | | | | | | | | | | | | | | probabilidade > 22.622: yes (10.0)
| | | | | | | | | | | | | | | | probabilidade > 23.558: no (21.0)
| | | | | | | | | | | | | | | probabilidade > 27.778
| | | | | | | | | | | | | | | | probabilidade <= 35.926: yes (10.0)
| | | | | | | | | | | | | | | | probabilidade > 35.926
| | | | | | | | | | | | | | | | | LCSR <= 0.4
| | | | | | | | | | | | | | | | | | probabilidade <= 37.969: yes (6.0)
| | | | | | | | | | | | | | | | | | probabilidade > 37.969: no (5.0)
| | | | | | | | | | | | | | | | | LCSR > 0.4: no (5.0)
| | | | | | | | | | | | | | | LCSR > 0.5: yes (5.0)
| | | | | | | | | | | | | | LCSR > 0.545: no (5.0)
| | | | | | | | | | | | | probabilidade > 39.432
| | | | | | | | | | | | | | probabilidade <= 61.782: yes (20.0)
| | | | | | | | | | | | | | probabilidade > 61.782: no (5.0)
| | | | | probabilidade > 64.946
| | | | | | LCSR <= 0.214: yes (32.0)
| | | | | | LCSR > 0.214
| | | | | | | LCSR <= 0.286: no (26.0)
| | | | | | | LCSR > 0.286
| | | | | | | | LCSR <= 0.444: yes (21.0)
| | | | | | | | LCSR > 0.444: no (5.0)
| | | | LCSR > 0.571
| | | | | LCSR <= 0.625: no (31.0)
| | | | | LCSR > 0.625
| | | | | | LCSR <= 0.636: yes (10.0)
| | | | | | LCSR > 0.636
| | | | | | | probabilidade <= 23.558: yes (5.0)
| | | | | | | probabilidade > 23.558
| | | | | | | | LCSR <= 0.75: no (20.0)
| | | | | | | | LCSR > 0.75
| | | | | | | | | LCSR <= 0.8: yes (5.0)
| | | | | | | | | LCSR > 0.8: no (11.0)
```