Desambiguação lexical de sentidos para o
português por meio de uma abordagem
multilíngue mono e multidocumento

Fernando Antônio Asevedo Nóbrega

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura:_____

Desambiguação lexical de sentidos para o português por meio de uma abordagem multilíngue mono e multidocumento

Fernando Antônio Asevedo Nóbrega

Orientador: Prof. Dr. Thiago Alexandre Salgueiro Pardo

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

USP – São Carlos
Julho de 2013

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

N754d Nóbrega, Fernando Antônio Asevedo
Desambiguação lexical de sentidos para o português
por meio de uma abordagem multilíngue mono e
multidocumento / Fernando Antônio Asevedo Nóbrega;
orientador Thiago Alexandre Salgueiro Pardo. -- São
Carlos, 2013.
106 p.

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

1. Desambiguação Lexical de Sentido. 2. WordNet.
3. Cenário Multidocumento. 4. Processamento de Língua
Natural. I. Pardo, Thiago Alexandre Salgueiro,
orient. II. Título.

Agradecimentos

Agradeço a toda minha família, que, mesmo com a distância, foi sempre presente e me apoiou em toda a caminhada. Em especial, agradeço a meus pais, Francisco e Maria Salete, meus irmãos presentes, Francisco Jr. e Sidnei juntamente com sua esposa Milane, e meus padrinhos, Jonas e Maria, que, em suas personalidades distintas, sempre me deram palavras de motivação, carinho e conforto.

Agradeço a todos meus amigos, colegas, companheiros sonhadores de ideias e empreendimentos, companheiros de república (e agregados), amigos do NILC e do grupo de Sumarização Automática. Todos vocês foram fontes de ideias, risadas, debates e, sem dúvida, hoje, são também minha família. Em especial, agradeço a Fernanda Bontempo, que compartilhou toda minha trajetória, em momentos de risadas ou tristezas, e sempre me motivou, ouviu e proporcionou momentos de descontração.

Agradeço também a todos os meus professores, que sem dúvida, contribuíram para minhas decisões e caminhadas. Em especial, agradeço ao Prof. Thiago Pardo, por sua orientação e direcionamento neste projeto. Todos vocês contribuíram em minha formação acadêmica, profissional e pessoal.

Agradeço também ao CNPq, pela bolsa de mestrado, CAPES e Fapesp, que indiretamente financiaram o andamento deste projeto. Da mesma forma, agradeço ao ICMC-USP e ao NILC pela infraestrutura (técnica e humana).

Agradeço também a Deus, pois, além de ser fonte de paz e força, me proporcionou conhecer as pessoas que agradeço e que contribuíram, de alguma forma, com este projeto.

“Mesmo se você falhar em meio a algo grandioso, é muito difícil falhar por completo.”

(Larry Page)

Resumo

A ambiguidade lexical é considerada uma das principais barreiras para melhoria de aplicações do Processamento de Língua Natural (PLN). Neste contexto, tem-se a área de Desambiguação Lexical de Sentido (DLS), cujo objetivo é desenvolver e avaliar métodos que determinem o sentido correto de uma palavra em um determinado contexto por meio de um conjunto finito de possíveis significados. A DLS é empregada, principalmente, no intuito de prover recursos e ferramentas para diminuir problemas de ambiguidade e, conseqüentemente, contribuir para melhorias de resultados em outras áreas do PLN. Para o Português do Brasil, pouco se tem pesquisado nesta área, havendo alguns trabalhos bem específicos de domínio. Outro fator importante é que diversas áreas do PLN engajam-se no cenário multidocumento, onde a computação é efetuada sobre uma coleção de textos, todavia, não há relato de trabalhos de DLS direcionados a este cenário, tampouco experimentos de desambiguação neste domínio. Portanto, neste trabalho de mestrado, objetivou-se o desenvolvimento de métodos de DLS de domínio geral voltado à língua Portuguesa do Brasil e o desenvolvimento de algoritmos de desambiguação que façam uso de informações multidocumento, bem como a experimentação e avaliação destes no cenário multidocumento. Para tanto, a fim de subsidiar experimentos, desenvolvimento e avaliação deste projeto, anotou-se manualmente o corpus CSTNews, caracterizado como um corpus multidocumento, utilizando a WordNet de Princeton como repositório de sentidos, que organiza os significados por meio de conjuntos de sinônimos (*synsets*) e relações linguísticas entre estes. Foram desenvolvidos quatro métodos de DLS e algumas variações, sendo: um método heurístico (para aferir valores de *baseline*); variações do algoritmo de [Lesk \(1986\)](#); adaptação do algoritmo de [Mihalcea and Moldovan \(1999\)](#); e uma variação do método de Lesk para o cenário multidocumento. Foram realizados três experimentos para avaliação dos métodos, cujos objetivos foram: determinar o desempenho geral dos algoritmos em todo o corpus; avaliar a qualidade de desambiguação de palavras mais ambíguas no corpus; e verificar o ganho de qualidade da desambiguação ao empregar informação multidocumento. Após estes experimentos, pôde-se observar que o método heurístico apresenta um melhor resultado geral. Contudo, é importante ressaltar que a maioria das palavras anotadas no corpus tiveram apenas um *synset*, que, normalmente, era o mais frequente, o que, conseqüentemente, apresenta um cenário mais propício ao método heurístico. Outro fato importante foi que, neste cenário, a diferença de desempenho entre o método de DLS multidocumento e o heurístico é estatisticamente irrelevante. Já para a desambiguação de palavras mais ambíguas, o método heurístico foi inferior, evidenciando que, para a desambiguação de palavras mais ambíguas, são necessários métodos mais sofisticados de DLS. Por fim, verificou-se que a utilização de informação multidocumento auxilia o processo de desambiguação. As contribuições deste trabalho podem ser agrupadas entre teóricas e técnicas. Nas teóricas, tem-se a investigação e análises da DLS no cenário multidocumento. Entre as contribuições técnicas, foram desenvolvidos métodos de DLS, um corpus anotado e uma ferramenta de anotação direcionados à língua Portuguesa do Brasil, que podem avançar as pesquisas em DLS para o idioma.

Palavras Chaves: Desambiguação Lexical de Sentido, WordNet, Cenário Multidocumento.

Abstract

The lexical ambiguity is considered one of the main barriers to improving applications of Natural Language Processing (NLP). In this context, it has been the area of Word Sense Disambiguation (WSD), whose goal is to develop and evaluate methods to determine the correct sense of a word in a given context by a finite set of possible meanings. The DLS is used mainly in order to provide resources and tools to reduce problems of ambiguity and thus contribute to improved results in other areas of NLP. In the Portuguese of Brazil, little has been researched in this area, with some work and specific domain. Another important factor is that many areas of NLP commit themselves in multidocument scenario, where the computation is performed on a collection of texts, however, there is no report of WSD work directed to this scenario, either disambiguation experiments in this field. Therefore, this master thesis aimed to develop methods of WSD general domain facing the Portuguese language in Brazil and the development of algorithms that make use of disambiguation multidocument informations, as well as experimentation and evaluation of the multidocument scenario. Therefore, in order to support experiments, development and evaluation of this project, the corpus CSTNews with 50 document collections, was manually annotated by means of synsets of the WordNet Princeton. Four methods were developed: A heuristic method (to measure values for baseline); variations of the [Lesk \(1986\)](#) algorithm; a adaptation of the [Mihalcea and Moldovan \(1999\)](#) algorithm; and a variation of the Lesk method for multidocument scenario. Three experiments were conducted to evaluate the methods, whose objectives were to determine the general performance algorithms across the corpus; evaluate the quality of disambiguation of most ambiguous words in the corpus, and check the gain quality of disambiguation by employing information multidocumento. After these experiments, it was observed that the heuristic method presents a better overall result. However, it is important to note that most of the words in the annotated corpus had only one synset, which usually was the most frequent, which, in turn, presents a scenario more conducive to the heuristic method. Another important fact was that in this scenario, the performance difference between the heuristic method and multidocument algorithm was statistically irrelevant. As for the disambiguation of most ambiguous words, the heuristic method was lower, indicating that, for the disambiguation of ambiguous words, more sophisticated WSD methods are needed. Finally, it has been found that the use of multidocument information assists the disambiguation process. The contributions of this work can be divided between theoretical and technical. In theory, there is the research and analysis of WSD in multidocument scenario. Among the techniques contributions, WSD methods have been developed an annotated corpus and annotation tool targeted to the Portuguese language in Brazil that can advance research in WSD for the language.

Keywords: Word Sense Disambiguation, WordNet, Multidocument Context

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Quadros	v
Lista de Tabelas	vii
Lista de Abreviações	ix
1 Introdução	1
1.1 Contexto	1
1.2 Objetivos	5
1.3 Hipóteses	9
1.4 Estrutura do Documento	10
2 Fundamentos	11
2.1 Ambiguidade Lexical	11
2.2 Desambiguação Lexical de Sentido	12
2.3 Avaliação de Sistemas em DLS	16
2.4 Recursos e Ferramentas Utilizadas em DLS	18
2.4.1 Recursos	18
2.4.2 Ferramentas	22
3 Trabalhos Relacionados	25
3.1 Abordagens Baseadas em Conhecimento	25
3.2 Abordagens Baseadas em Córpus	38
3.3 Síntese dos Trabalhos Discutidos	44
4 Anotação de Córpus	49
4.1 Anotação do Córpus	49
4.1.1 Metodologia de Anotação	51
4.1.2 Ferramenta de Anotação: NASP	53
4.2 Avaliação de Concordância	59
4.3 Resultado da Anotação	61
5 Métodos de DLS Investigados	65
5.1 Pré-processamento e Recuperação de <i>Synsets</i>	66
5.2 Método Heurístico	67

5.3	Método de Lesk	67
5.4	Método Irrestrito de Mihalcea	69
5.5	Desambiguação com Grafo de Co-ocorrência Multidocumento	71
5.6	Avaliação	73
6	Conclusões	79
6.1	Conclusão sobre Anotação do Córpus	79
6.2	Conclusão sobre os Métodos de DLS Desenvolvidos	80
6.3	Contribuições	82
6.4	Trabalhos Futuros	82
	Referências Bibliográficas	85
A	Apêndice: Textos de Treinamento	91
A.1	Estadão	91
A.2	Folha de São Paulo	92
B	Apêndice: Resumo de Anotação	95

Lista de Figuras

1.1	Níveis de Complexidade em PLN	2
1.2	Exemplo de DLS por meio da abordagem multilíngue.	7
3.1	Trabalhos de DLS	47
4.1	Percentual de ocorrência das etiquetas morfossintáticas no corpus CST-News (Somente classes gramaticais abertas)	50
4.2	Metodologia de Anotação	53
4.3	NASP (Nilc – Anotador de Sentidos para o Português)	54
4.4	Exemplo da NASP: anotando uma palavra.	56
4.5	Exemplo da NASP: selecionando uma palavra pré-anotada.	57
4.6	Hierarquia de Etiquetas XML da Ferramenta NASP	58
4.7	Distribuição de <i>synsets</i> por palavra no corpus	61
4.8	Distribuição de <i>synset</i> por palavra em coleção	62
4.9	Distribuição de possíveis <i>synsets</i> ordenados	62
4.10	Distribuição de possíveis <i>synsets</i> ordenados por tradução	63
4.11	Distribuição de Possíveis <i>Synsets</i> Ordenados por Frequência	64
5.1	Exemplo de Grafo de Co-ocorrência	72
5.2	Exemplo de Grafo de Co-ocorrência para Palavra “avião” por meio do Cluster de número 3	73
5.3	Relação de Palavras Desambiguadas pelos Métodos	77

Lista de Quadros

1.1Definição do verbo bancar encontrado na WordNet-BR	3
3.1Trabalhos de DLS	46
4.1Palavras com discordância total entre os anotadores	61
4.2Exemplo de Palavras com Respectivas Possibilidades de Desambiguação	64
5.1Sentença Exemplo e Atributos da Palavra “avião”	66

Lista de Tabelas

4.1	Valores de concordância	60
5.1	Análise da Variedade de Rótulos da Adaptação do Algoritmo de Lesk (1986)	68
5.2	Avaliação <i>all-words</i>	74
5.3	Avaliação <i>lexical-sample</i>	75

Lista de Abreviações

AM	Aprendizado de Máquina
CST	<i>Cross-document Structure Theory</i>
CV	Caixeiro Viajante
DLS	Desambiguação Lexical de Sentido
IA	Inteligência Artificial
IDC	<i>International Data Corporation</i>
NASP	Nilc – Anotador de Sentidos para o Português
PA	Palavra Alvo
PLN	Processamento de Língua Natural
PT-BR	Português do Brasil
RPS	Rotulagem de Papéis Semânticos
RS	<i>Repositório de Sentidos</i>
RST	<i>Rhetorical Structure Theory</i>
Wn-BR	WordNet-BR
Wn-Pr	WordNet de Princeton
WNSR	<i>WordNet::SenseRelate</i>
WSD	<i>Word Sense Disambiguation</i>

Introdução

1.1 Contexto

Diversas áreas de pesquisa desenvolvem aplicações que necessitam processar arquivos, documentos ou textos descritos em língua natural. Como exemplo, podem-se citar aplicações diretas ou indiretamente relacionadas ao Processamento de Língua Natural (PLN), como métodos de agrupamento de documentos, auxílio à escrita, recuperação de informação, simplificação textual, sumarização automática, tradução automática, dentre outros. Essas aplicações são impulsionadas pela grande quantidade de informação textual disponível, principalmente intermediada pela Web. O crescimento dessa quantidade de informação é apontado por [Gantz and Reinsel \(2011\)](#) em artigo publicado pela IDC (*International Data Corporation*). Os autores estimam uma produção de 1.8 *zettabytes* de informação Web em 2011, que corresponde a um crescimento de 9 vezes mais informação em relação aos 5 anos anteriores.

Neste cenário, aplicações do PLN podem empregar diversas abordagens, que podem ser classificadas entre superficiais ou profundas, quando, respectivamente, fazem uso de pouca ou muita informação linguística. Geralmente, essas informações linguísticas são discretizadas em diferentes níveis de conhecimento. [Jurafsky and Martin \(2009\)](#), por exemplo, sugerem os níveis da fonética/fonologia, morfologia, sintaxe, semântica e pragmática/discurso. Na Figura 1.1, pode-se observar a disposição desses níveis de conhecimento ordenados pela grau de subjetividade/abstração da informação, ou seja, os níveis mais altos são considerados mais subjetivos que os níveis inferiores. Com essa classificação, pode-se classificar como abordagens superficiais aquelas que utilizam informações

dos níveis inferiores e, analogamente, abordagens profundas aquelas que empregam conhecimentos dos níveis mais altos¹.

Figura 1.1: Níveis de Complexidade em PLN

Em geral, métodos de PLN baseados em abordagens superficiais demandam menos tempo para serem desenvolvidos. Contudo, características inerentes à língua natural podem dificultar o processamento “superficial” da informação textual. Por exemplo, a sentença “o banco quebrou” pode apresentar diferentes significados devido às possibilidades de sentidos da palavra “banco”, tais como: um banco de madeira (móvel) quebrou, uma instituição financeira faliu, um *software* de gerenciamento de dados parou de funcionar, etc. O uso dessa sentença em buscadores de informação baseado em abordagens superficiais, por exemplo, pode retornar resultados irrelevantes aos usuários. Isso ocorre porque métodos superficiais não consideram os sentidos das palavras, pois não usam conhecimento semântico, que é considerado conhecimento profundo.

O problema supracitado caracteriza-se pela existência da ambiguidade lexical. Este fenômeno linguístico ocorre quando uma palavra pode assumir diferentes significados dependendo do contexto em que é empregada (Piruzelli and Dias da Silva, 2010). Essa característica da língua natural é tratada no nível semântico, portanto, classificado como conhecimento profundo.

Neste contexto, encontra-se a área de Desambiguação Lexical de Sentido (DLS, do inglês *Word Sense Disambiguation* – WSD), cujo objetivo é desenvolver métodos que determinem o significado mais adequado de uma palavra, em um contexto especificado, por meio de um Repositório de Sentidos (RS) (Agirre and Edmonds, 2006). Este campo da ciência surgiu motivado pela área de Tradução Automática, visto a necessidade de determinar o sentido da palavra para, posteriormente, traduzi-la. Por exemplo, assim como anteriormente apresentado, a palavra banco pode assumir os sentidos de um móvel de madeira ou uma instituição financeira. No primeiro caso, a tradução “seat” poderia ser

¹Ressalta-se que não há um consenso na separação dos níveis que são tratados pelas abordagens superficiais e profundas. Contudo, neste trabalho, entende-se que conhecimento semântico e discursivo são conhecimentos profundos.

utilizada, porém, no segundo caso, a tradução mais adequada seria “bank”.

Os RS são estruturas computacionalmente tratáveis (no âmbito da DLS) compostas por conjuntos formados de, pelo menos, pares de palavras e respectivos sentidos, tais como dicionários, ontologias, *thesaurus*, *wordnets*, etc. A opção por determinado RS deve respaldar-se nas necessidades do método de DLS proposto, por exemplo, um método de desambiguação geográfica (cujo objetivo é desambiguar expressões como “São Paulo”, que pode referenciar a cidade ou estado) requer RS específicos deste domínio. Por outro lado, alguns trabalhos abrangem repositórios mais genéricos, como o de Lesk (1986), que apresenta um método de DLS genérico por meio do uso de dicionários como RS. Contudo, Miller (1995) salienta que alguns RS, tais como dicionários, apesar de serem computacionalmente manipuláveis, são estruturados para utilização humana, e, conseqüentemente, são menos efetivos à DLS.

Repositórios frequentemente utilizados e com estruturas mais elaboradas são as *wordnets*. Estes RS são estruturados com base em conjuntos de sinônimos, denominados *synsets*, e as relações linguísticas entre estes. Cada *synset* representa um possível sentido e possui as seguintes informações: (1) conjunto de palavras sinônimas, agrupadas também pela classe gramatical; (2) glosa, que é uma descrição informal do sentido do *synset*; (3) conjunto de exemplos, composto por sentenças que empregam palavras do item 1; e (4) relações linguísticas com outros *synsets*. Ressalta-se que alguns *synsets* podem não conter algumas informações supracitadas. Por exemplo, alguns *synsets* não possuem exemplos.

A primeira *wordnet*, e também a mais difundida, é a WordNet de Princeton² (Wn-Pr) (Fellbaum, 1998), construída para o idioma Inglês. Para o Português do Brasil (PT-BR), tem-se a WordNet-BR³ (Wn-BR) (Dias da Silva, 2005), ainda em fase de desenvolvimento durante o andamento deste trabalho.

Quadro 1.1: Definição do verbo bancar encontrado na WordNet-BR

Sentido 1	
Synset:	afetar, aparentar, bancar, dissimular, encenar, fazer, fingir, posar, simular
Significado:	Criar uma pretensão [de]
Sentido 2	
Synset:	bancar, custear, financiar
Significado:	Prover a(s) despesa(s) para a construção ou realização de algo

Para exemplificar o uso de *synsets*, é apresentada no Quadro 1.1 a definição do verbo “bancar” encontrada na Wn-BR. Pode ser observado que: (1) os dois *synsets* diferem quanto ao significado; e (2) todas as palavras do primeiro *synset* podem assumir o sentido “cria uma pretensão [de]” e, de forma análoga, todas as palavras do segundo *synset* podem

²<http://wordnet.princeton.edu/>

³<http://www.nilc.icmc.usp.br/wordnet/>

assumir o sentido de “prover a(s) despesa(s) para a construção ou realização de algo”. Nesse exemplo, na sentença “A instituição irá bancar os custos da viagem”, o verbo “bancar” possui o Sentido 2 da Wn-BR, podendo, inclusive, ser substituído pelos seus respectivos sinônimos “custear” ou “financiar” incluídos no conjunto de sinônimos.

A DLS é empregada, principalmente, em etapas de pré-processamento de outras aplicações, visando diminuir os casos de ambiguidade e contribuir para melhorias nos resultados destas áreas de pesquisa (Mitkov, 2004). Normalmente, o processo de desambiguação ocorre individualmente para cada palavra em seu respectivo contexto local, representado, normalmente, somente pelas palavras em torno da Palavra Alvo (PA – palavra a ser desambiguada). Em contraste a este cenário, onde a DLS é computada localmente, diversas aplicações inserem-se no cenário multidocumento, onde a computação é realizada em um conjunto de documentos, não mais em textos isolados ou sentenças. Como exemplo, têm-se as áreas de: Agrupamento de Textos; Recuperação de Informação; Sumarização Automática Multidocumento; etc. Consequentemente, ao empregar métodos de DLS habituais, que utilizam somente informações de contextos locais, as informações multidocumentos, que podem auxiliar na desambiguação, ficam desperdiçadas.

Alguns trabalhos encontrados na literatura relatam métodos de DLS capazes de usufruir de informações do contexto multidocumento. Por exemplo, Agirre and Soroa (2009) apresentam métodos de desambiguação baseados em algoritmos de ranqueamento por meio de grafos. Tais métodos criam relações de todas as palavras do contexto (mono ou multidocumento) para seus respectivos possíveis *synsets* em um grafo da Wn-Pr e, posteriormente, empregam alguns algoritmos da literatura (bem como algumas variações) para encontrar os sentidos mais importantes (com maior valor de relevância, dada a rede de relacionamentos no grafo) no contexto. Entretanto, não há relatos de avaliação, desenvolvimento e experimentações destes métodos diretamente no contexto multidocumento, tampouco há estudos do “comportamento” dos sentidos das palavras neste domínio.

Ainda no cenário da DLS, apesar de grandes esforços estarem sendo realizados para o término da Wn-BR, pouco se tem investigado a respeito do desenvolvimento e avaliação de métodos de desambiguação direcionados ao Português do Brasil (PT-BR). Neste cenário, podem-se citar os trabalhos de Specia (2007) e Machado et al. (2011).

O trabalho de Specia (2007) destina-se à tarefa de DLS direcionada à tradução automática entre Português e Inglês, cujo objetivo é a desambiguação de 10 verbos do Inglês, que são considerados mais ambíguos. Já Machado et al. (2011) apresentam métodos de desambiguação no contexto geográfico, ou seja, desambiguar nomes de cidades, estados, regiões, etc. Contudo, tem-se que estes trabalhos, mesmo abordando o idioma Português, são projetos bem específicos em suas áreas de atuação e, consequentemente, são pouco efetivos a diversas outras áreas do PLN destinada ao PT-BR.

Nas seções subsequentes deste capítulo, serão apresentados os objetivos de trabalho na Seção 1.2 e as hipóteses de pesquisa na Seção 1.3. E, por fim, na Seção 1.4, descreve-se a estrutura utilizada na redação desta dissertação de mestrado.

1.2 Objetivos

Métodos de DLS podem auxiliar diversas aplicações inseridas em PLN. Contudo, para o PT-BR, encontram-se poucas pesquisas nesta linha. Portanto, tem-se que o principal objetivo deste trabalho de mestrado foi investigar, desenvolver e avaliar alguns métodos de DLS de uso geral para o Português do Brasil.

Como dito anteriormente, de modo geral, poucos trabalhos de DLS engajam-se no cenário multidocumento. Portanto, o segundo objetivo deste trabalho foi avaliar a eficácia da utilização de características multidocumento em métodos de DLS, bem como o comportamento da distribuição das palavras e respectivos sentidos neste cenário.

No intuito de prover recursos e ferramentas de desambiguação para aplicações de PLN, com ênfase no PT-BR, optou-se pela investigação de métodos de DLS por meio da abordagem baseada em conhecimento. Métodos desta abordagem, apesar de apresentarem resultados inferiores aos métodos de Aprendizado de Máquina (AM), são caracterizados por não restringir o vocabulário, e, conseqüentemente, podem ser adotados em cenários variados.

A fim de limitar o escopo da pesquisa, objetivou-se a desambiguação apenas de palavras da classe morfossintática de substantivo comum. Esta restrição pautou-se em investigação por meio de *cópus*, na qual se observou que palavras desta classe são mais frequentes em um mesmo texto, e em trabalhos da literatura, como Plaza and Diaz (2011), que relatam que a desambiguação de substantivos comuns contribui mais efetivamente para melhores resultados em algumas aplicações do PLN.

Considerando os objetivos supracitados, utilizou-se a Wn-Pr como RS. Assim, este trabalho caracteriza-se como uma abordagem multilíngue, visto a utilização de recursos da língua inglesa para investigação de métodos de DLS com ênfase no PT-BR. As principais razões para utilizar a Wn-Pr em vez da Wn-BR foram:

1. a Wn-Pr é a mais difundida e utilizada na literatura. Mesmo sendo desenvolvida para o idioma Inglês, podem-se utilizar tradutores automáticos Português-Inglês e anotação de *cópus* para viabilizar os métodos desenvolvidos para o Português-BR;
2. a Wn-BR, no presente momento deste trabalho, encontra-se ainda em desenvolvimento;

3. a Wn-BR possui um mapeamento de seus *synsets* para seus representantes da Wn-Pr. Assim, os métodos desenvolvidos neste trabalho podem futuramente ser mapeados para a versão do Português brasileiro. Este mapeamento é possível, pois *wordnets*, em sua estruturação, possuem, para cada *synset*, identificadores numéricos únicos (*synset-id*) que são empregados para relacionar *wordnets* de diferentes idiomas;
4. o TEP 2.0 (Maziero et al., 2008) é um *thesaurus* desenvolvido para a língua portuguesa. Contudo, a estruturação deste RS é constituída por uma hierarquia mais simples entre as palavras (apenas sinonímia e antonímia) do que a Wn-Pr. Outra característica importante é que não possui apontadores para outras *wordnets*. Portanto, a utilização do recurso supracitado pode ser menos efetiva a outras aplicações do PLN direcionadas ao PT-BR.
5. a Onto.PT (Oliveira and Gomes, 2012), considerada uma ontologia lexical para o Português, ao contrário do TEP 2.0, possui sentidos indexados à Wn-Pr. Contudo, trata-se de um recurso recente e gerado semiautomaticamente por meio de usos de dicionários e *thesaurus* (incluindo o TEP 2.0); e
6. Outros recursos, como a EuroWordNet (Vossen, 1998), que é uma *wordnet* para diversos idiomas europeus, são pagos (Bond and Paik, 2012).

Um exemplo da proposta multilíngue de DLS presente neste trabalho é mostrado na Figura 1.2. Neste exemplo, é apresentada a desambiguação da palavra “avião” empregada na sentença “...o avião apresentava problemas no reversor...” extraída do corpús CSTNews (Aleixo and Pardo, 2008; Cardoso et al., 2011). Primeiramente, é realizada a tradução⁴ da PA, e com estas traduções listam-se todos os respectivos *synsets* da Wn-Pr (aqui, referenciados somente pelo conjunto de sinônimos e ordenados segundo a listagem da Wn-Pr). Após a indexação das possibilidades de sentidos para a PA, aplica-se o algoritmo de desambiguação para determinar o *synset* adequado, que na figura é referenciado por uma seta contínua.

Para atingir estes objetivos, torna-se necessária a utilização de um corpús anotado para avaliação dos métodos desenvolvidos. Neste trabalho, utilizou-se o corpús CSTNews (Aleixo and Pardo, 2008; Cardoso et al., 2011) (que será abordado com mais detalhes na Seção 2.4.1), que se trata de um corpús bastante utilizado em aplicação de Sumarização Automática. Outra característica deste recurso, que tornou pertinente sua utilização, é o gênero textual de seus textos. O CSTNews aborda o gênero jornalístico, que é representado em uma linguagem mais comum, pois é destinado a um grande público com

⁴Tradução realizada por meio do dicionário bilíngue WordReference®, que será melhor detalhado na Seção 2.4.

Figura 1.2: Exemplo de DLS por meio da abordagem multilíngue.

diversidade de cultura, escolaridade, etc. Assim, acredita-se que o uso desse gênero textual permitirá que os métodos de DLS desenvolvidos abranjam diferentes tipos de textos, principalmente na Web.

Ressalta-se que, anteriormente a este trabalho, o corpus CSTNews não era semanticamente anotado com sentidos, ou seja, o corpus não dispunha de um conjunto de palavras desambiguadas por meio de *synsets* ou outra especificação de significados (entradas de um dicionário, por exemplo). Portanto, durante este trabalho de mestrado, foi necessária uma etapa para anotação semântica do CSTNews (mais detalhada no Capítulo 4).

Para atingir os objetivos de pesquisa supracitados, enumeram-se as seguintes principais etapas de projeto:

1. adicionar informação semântica ao corpus CSTNews, desambiguando cada substantivo comum por meio de sua indexação ao *synset* mais adequado na Wn-Pr;
2. investigar, desenvolver ou adaptar alguns métodos de DLS, monodocumento e multidocumento, para o Português do Brasil; e
3. avaliar os métodos de DLS desenvolvidos.

Neste trabalho, foram selecionados três métodos de DLS encontrados na literatura. O primeiro, frequentemente empregado para aferir valores de *baseline*, baseia-se no método heurístico de adotar o sentido mais frequente para uma palavra (Mihalcea, 2006). Devido à característica multilíngue deste trabalho, que necessita de etapas de tradução, adotou-se a tradução mais frequente de uma palavra no PT-BR para inglês e respectivo *synset* mais frequente. O segundo método selecionado foi o algoritmo de Lesk (1986), que, por meio de dicionários, determina o significado de uma palavra por meio do cálculo da similaridade

de seus possíveis sentidos com características das palavras no contexto. Assim, por este algoritmo, é possível desambiguar qualquer palavra que esteja listada em um dicionário. Especificamente neste trabalho, adotou-se como referência a adaptação do algoritmo de Lesk para utilização da Wn-Pr apresentada por [Banerjee \(2002\)](#). Ainda para o algoritmo de Lesk, investigou-se também uma versão simplificada do método apresentado por [Kilgarriff et al. \(2000\)](#), cujo objetivo foi diminuir a complexidade computacional da versão original do algoritmo. A fim de investigar métodos de DLS no contexto multidocumento, esta variação do algoritmo de Lesk também foi adaptada para este cenário. O terceiro método selecionado foi o algoritmo de [Mihalcea and Moldovan \(1999\)](#). Este método realiza a desambiguação por meio de ferramentas de recuperação de informação usando pares de palavras para construção de *queries* (sentenças de busca para sistemas de recuperação de informação). Este método, assim com o algoritmo de Lesk, é caracterizado por ser irrestrito.

Os métodos de DLS supracitados foram selecionados por suas características principais, das quais se podem citar: 1) bastante difundidos na literatura; 2) algoritmos que permitem a desambiguação de qualquer palavra que esteja listada em um RS, o que não restringe o domínio textual utilizado; e 3) bastante utilizados para aferir similaridades entre palavras, o que, conseqüentemente, pode contribuir para aplicações do PLN que necessitam de tais métricas.

Os métodos foram desenvolvidos e, posteriormente, analisados por três óticas. Na primeira, foram comparados os desempenhos dos métodos com o valor *baseline*, tradução e respectivo *synset* mais frequentes, por meio de todas as instâncias anotadas no *corpus*. Esta análise inicial, por meio de um contexto geral, pautou-se em investigar a viabilidade e o desempenho de métodos de DLS mais sofisticados (em relação à abordagem heurística) compostos por uma abordagem multilíngue.

A segunda investigação objetivou avaliar o desempenho dos métodos na desambiguação de um conjunto de palavras, consideradas mais ambíguas por apresentarem diferentes sentidos anotados no *corpus*. Nesse cenário, em contraste à primeira análise supracitada, os métodos desenvolvidos apresentaram-se superiores ao método *baseline*.

Na terceira análise, por meio das duas óticas anteriormente citadas, investigou-se o ganho de desempenho por meio da utilização de informação multidocumento no processo de desambiguação. Assim, compararam-se, mais detalhadamente, os métodos desenvolvidos em relação ao algoritmo de Lesk adaptado para o cenário multidocumento, desenvolvido neste trabalho. Pôde-se observar que a informação multidocumento contribui positivamente para a DLS.

1.3 Hipóteses

Diversos métodos de DLS vêm sendo desenvolvidos, porém, poucas pesquisas são destinadas ao idioma PT-BR. Talvez, a principal razão desta escassez de pesquisas em DLS seja a falta de recursos semânticos refinados para o idioma. Porém, dada a possibilidade de utilização de tradutores bilíngues, anotação de *cópus* e aplicação de alguns métodos de desambiguação independentes de língua, tem-se como hipótese geral deste trabalho que: é possível desenvolver métodos de DLS para o Português do Brasil utilizando a Wn-Pr como repositório de sentidos, de tal forma que seus resultados sejam superiores ao método *baseline* (mais detalhado na Seção 5.2), que consiste em atribuir o sentido mais frequente.

Ao comparar o desempenho dos métodos desenvolvidos em relação ao método *baseline*, observou-se que, de modo geral, o método heurístico (sentido mais frequente) apresentou maior precisão. Contudo, um dos métodos, que emprega informação multidocumento, mostrou-se inferior em apenas 0,4 pontos percentuais. Ressalta-se que, após verificar a distribuição de sentidos por palavra anotada manualmente no *cópus* (detalhada na Seção 4.3), tem-se que a maioria dessas palavras apresentam apenas um significado anotado e que, normalmente, corresponde ao mais frequente, conseqüentemente, proporcionando um cenário mais promissor ao uso do método heurístico empregado.

Outra hipótese deste trabalho é que a utilização da Wn-Pr, desenvolvida para a língua inglesa, pode ser suficiente para o desenvolvimento dos métodos de DLS para o Português, mesmo na possibilidade de lacunas lexicais, que são ausências ou especificações de sentidos entre diferentes línguas. Neste caso, há necessidade de generalização, quando uma palavra do português não possui correspondente no inglês, e de especificação, quando uma palavra no português possui várias possibilidades no inglês. Por exemplo, o conceito “caipirinha”, típica bebida brasileira, não possui correspondência direta com a Wn-Pr, o que torna necessário generalizar o termo “caipirinha” para o conceito de “bebida”. Por outro lado, a palavra “dedo” que, no PT-BR, representa tanto dedos das mãos quanto dos pés, é respectivamente representada no inglês por “*finger*” e “*toe*”.

No caso de generalização, apresentado no primeiro exemplo anterior, torna-se necessária a utilização de processos que, de certa forma, identifiquem o conceito mais geral que uma palavra. No processo de anotação manual do *cópus*, este processo, sempre que necessário, foi realizado pelos participantes. Já a especialização de conceitos, caracteriza-se como um processo inerente à desambiguação.

A hipótese anterior pôde ser observada por meio da avaliação da abrangência dos métodos de DLS, que corresponde ao percentual de palavras do conjunto de teste que os métodos conseguem desambiguar (correta ou incorretamente). Tem-se que, em geral, a abrangência dos métodos alcança 91,10%. Ou seja, apesar da utilização de uma abor-

dagem multilíngue, os métodos propostos não foram capazes de desambiguar somente poucas palavras, apenas 8,90% das instâncias do *corpus*.

Partindo do objetivo de desenvolver métodos de DLS para contribuir com outras aplicações, que podem estar inseridas no cenário multidocumento, tem-se a hipótese que este cenário pode contribuir para a DLS, pois textos relacionados podem conter informações que podem fortalecer pistas do sentido correto de uma palavra. Por exemplo, informações complementares podem disponibilizar pistas não existentes em alguns textos da coleção; informações redundantes podem fortalecer as pistas já apresentadas em outros textos, etc.

Como dito na seção anterior, uma das análises dos métodos pautou-se em investigar o ganho da utilização multidocumento no processo de DLS. Assim, apesar de o método multidocumento desenvolvido apresentar restrições, como adotar apenas um sentido por palavra em uma mesma coleção de documentos, mostrou-se superior aos demais.

Considerando uma análise do *corpus*, na qual se computou a quantidade de sentidos diferentes atribuídos a uma palavra em uma mesma coleção (melhor detalhado na Seção 4.3), observou-se que as palavras tendem a manter o mesmo sentido na coleção, o que proporciona um cenário mais promissor ao método multidocumento desenvolvido. Contudo, não se pode afirmar que todos os cenários multidocumento apresentam a mesma distribuição, vista a dimensão do *corpus* utilizado.

1.4 Estrutura do Documento

O conteúdo desta dissertação de mestrado se organiza em 6 capítulos. No Capítulo 2, é apresentada a fundamentação teórica sobre ambiguidade lexical e a área de DLS, os métodos de avaliação de sistemas de desambiguação, bem como as ferramentas e recursos utilizados neste trabalho. No Capítulo 3, disserta-se sobre os trabalhos relacionados encontrados na literatura.

Os resultados obtidos neste trabalho são organizados em dois capítulos. No Capítulo 4 disserta-se sobre a anotação de *corpus* necessária para desenvolvimento e avaliação dos métodos de DLS, que são apresentados no Capítulo 5. Por fim, no Capítulo 6, são apresentadas a conclusão e a discussão final deste trabalho.

Fundamentos

Neste capítulo, é abordado o tema da Desambiguação Lexical de Sentido (DLS). Nesse intuito, conceitos linguísticos sobre ambiguidade são apresentados na Seção 2.1 a fim de fundamentar o objeto de estudo (fenômeno linguístico) da DLS. Na Seção 2.2, são abordadas as principais características, conceitos, dificuldades e história da DLS, com objetivo de apresentar uma visão deste ramo de pesquisa. Nesta mesma linha, formas de avaliação de métodos de DLS são abordadas na Seção 2.3. Por fim, na Seção 2.4, é disposto um conjunto de recursos e ferramentas encontrados na literatura para aplicações de DLS.

2.1 Ambiguidade Lexical

O fenômeno da ambiguidade se apresenta em diversas formas e níveis. Piruzelli and Dias da Silva (2010), por exemplo, apresentam a seguinte distinção entre tipos de ambiguidades: lexical, estrutural, anafórica (ou referencial) e temática.

A ambiguidade lexical ocorre principalmente pela existência dos fenômenos da homonímia e polissemia. A homonímia ocorre quando os sentidos atribuídos a uma palavra não possuem alguma relação semântica. Já o fenômeno da polissemia ocorre quando os sentidos atribuídos a uma palavra possuem relação semântica. Por exemplo, a palavra “banco” pode significar: (1) uma instituição financeira, (2) um banco de areia (depósito), (3) banco de sangue, etc. Todos esse sentidos possuem uma relação semântica, pois representam conceitos destinados ao armazenamento de algo: em 1, armazena-se dinheiro; em 2, deposita-se areia; e, em 3, armazena-se sangue. Assim, por estes significados apresen-

tarem relação semântica, esses termos são considerados polissêmicos. Por outro lado, a palavra “manga”, pode referenciar o conceito de: (a) uma fruta; (b) uma parte da roupa. Estes conceitos, em princípio, não possuem nenhuma relação semântica, portanto, são sentidos homônimos. Ressalta-se que a distinção entre esses fenômenos linguísticos é sutil. Por exemplo, a palavra “manga”, no sentido de “fruta” ou “manga de camisa”, não apresenta, a princípio, relação semântica entre esses possíveis significados (Piruzelli and Dias da Silva, 2010).

A ambiguidade estrutural se apresenta no nível sintático, quando há várias formas de se organizar as sentenças ou frases. Por exemplo, na sentença “O menino olhou a menina com o binóculo”, não se sabe exatamente quem está usando o binóculo, se é o menino que está usando o binóculo para observar a menina, ou se é a menina que está sendo observada enquanto usa o binóculo. Alguns casos deste tipo de ambiguidade podem ser solucionados por conhecimento de mundo, por exemplo, substituindo a palavra “menina” por “pássaro” tem-se a seguinte sentença: “o menino olhou o pássaro com o binóculo”. Assim, tendo o conhecimento que pássaros não utilizam objetos, sabe-se que o sentido da sentença é que o menino estava usando o binóculo para observar o pássaro, e não o pássaro que estava sendo observado enquanto usava o binóculo.

A ambiguidade anafórica está presente no uso de palavras que referenciam outros conceitos no texto ou sentença. Por exemplo, na sentença “Alguém jogou a xícara na mesa e a quebrou”, não se sabe precisamente qual foi o objeto quebrado, pois há uma ambiguidade de referência presente no artigo “a”. Novamente, aplicando-se conhecimento de mundo sabe-se que, provavelmente, o objeto quebrado foi a xícara, pois sua massa e rigidez são menores do que a da mesa. Normalmente, esse tipo de ambiguidade é abordada em uma área de pesquisa distinta, a Resolução de Anáforas.

A ambiguidade temática se apresenta na utilização de preposições que podem apresentar diferentes funções. Por exemplo, na sentença “Entregue o troféu para Ana”, a preposição “para” pode estar sendo utilizada para indicar que Ana ganhou o troféu e para simplesmente entregar o troféu para Ana, sem que este seja realmente seu.

Dadas as definições supracitadas de ambiguidade, ressalta-se que no escopo deste trabalho de mestrado foi tratada apenas a ambiguidade lexical. Portanto, as seções subsequentes deste capítulo irão abordar conceitos relacionados somente a este tipo de ambiguidade.

2.2 Desambiguação Lexical de Sentido

A Desambiguação Lexical de Sentido (DLS) consiste na tarefa de determinar o sentido mais adequado para uma palavra, dado o contexto em que esta foi empregada e um

conjunto finito de sentidos (RS). Pesquisas nesta área iniciaram-se por interesses em aplicações de Tradução Automática (TA), no intuito de melhorar a automação dessa tarefa por computadores ([Agirre and Edmonds, 2006](#)).

O problema da DLS é considerado um problema completo da Inteligência Artificial (IA), ou seja, trata-se de um problema que será resolvido apenas quando todos os outros grandes problemas da IA forem resolvidos, por exemplo, representar conhecimento humano que sempre evolui ([Ide and Véronis, 1998](#)). Essa classificação da tarefa da DLS pauta-se, principalmente, em seu dinamismo, pois novas palavras são criadas com o passar do tempo e, conseqüentemente, novos significados são atribuídos a estas. Por exemplo, o surgimento de novas ferramentas pode estimular a aparição de novas palavras, como o verbo *to google* do idioma Inglês, ou a versão em Português “googlar”, que representam o conceito de realizar buscas na Web.

Provavelmente, o primeiro método computacional de desambiguação foi proposto por [Lesk \(1986\)](#). Seu método se caracteriza pelo uso de dicionários e pela independência de língua, ou seja, seu método originalmente desenvolvido para o Inglês pode ser portado para outros idiomas, como o Português, por exemplo, por meio da alteração do dicionário utilizado. Este trabalho ainda é muito utilizado para aferir medidas de referência (valores *baseline*).

Com o surgimento de novos recursos para o PLN, tais como *corpus*, documentos Web e *wordnets*, dentre outros, novas abordagens para a DLS começaram a surgir. O impacto ocasionado, além do crescimento de pesquisas na área, foi o aumento do escopo da DLS, que passou a ser investigada em conjunto com outras áreas de pesquisa como Recuperação de Informação, Sumarização Automática, Sistemas de Perguntas e Respostas, etc. ([Agirre and Edmonds, 2006](#)). Esse contexto fez com que as tarefas realizadas por sistemas de DLS variassem conforme o domínio da aplicação. Contudo, há na literatura três tarefas principais para a DLS ([Jurafsky and Martin, 2009](#)). São elas:

- desambiguação de uma amostra de palavras (referenciada por *lexical sample*): os métodos de DLS devem desambiguar um conjunto pré-determinado de palavras, sendo também especificados seus possíveis sentidos;
- desambiguação de todas as palavras (referenciada por *all words*): os métodos de DLS devem desambiguar todas as palavras em um texto; e
- desambiguação por transferência: os métodos de DLS devem desambiguar uma palavra determinando sua melhor tradução em outro idioma.

As tarefas supracitadas ainda podem ser modificadas para algum domínio específico. Por exemplo, para algumas aplicações, pode ser útil desambiguar apenas palavras de uma

classe gramatical específica. [Plaza and Diaz \(2011\)](#), por exemplo, investigaram métodos de DLS por meio de grafos em sistemas de Sumarização Automática. Após realizarem alguns experimentos, relatam que a desambiguação de verbos não contribuiu para melhoria dos sumários. Por outro lado, os autores relatam que a desambiguação de apenas substantivos contribui mais efetivamente à Sumarização.

Outras características que diferenciam métodos de DLS são as informações, ou atributos, que estes usam para determinar os sentidos. [Lesk \(1986\)](#), por exemplo, faz uso apenas do conceito de Janela de Palavras para determinar o contexto da palavra alvo, mas não se preocupa com a classe gramatical. Na literatura relacionada, diversos atributos são citados. Porém, os mais utilizados são:

- *bag of words*: consiste nas palavras que estão em torno da PA; a quantidade de palavras a serem analisadas varia de acordo com o método a ser proposto;
- classe gramatical: consiste na classe gramatical da palavra alvo. Este conhecimento é importante, pois algumas palavras possuem diferentes sentidos conforme sua classe gramatical, por exemplo, o uso da palavra “banco” como substantivo refere-se a instituição financeira (dentre outros), e seu uso como verbo refere-se ao sentido de custear algo;
- similaridade semântica: consiste em valores numéricos que representam o quanto duas palavras são semanticamente relacionadas, ou seja, a proximidade entre os significados de duas palavras. Por exemplo, o conceito de porta é semanticamente mais relacionado ao conceito de casa do que ao conceito de estrada. Entretanto, há diversas formas de se calcular tal valor. De fato, a maioria dos métodos de DLS baseados em conhecimento baseia-se na aplicação de métricas de similaridade, admitindo que o sentido mais adequado é aquele que possui maior similaridade com o contexto da PA;
- contexto: informação utilizada para especificar o cenário em que a palavra alvo está sendo empregada. Este atributo pode ser subdividido em diversos níveis, porém, geralmente, é mais frequentemente especificado como contexto local, representando as palavras em torno da palavra alvo. Normalmente, para representar este atributo, a abordagem de *bag of words* é utilizada.

Geralmente, métodos de DLS empregam algoritmos com alta complexidade computacional. Portanto, a fim de prover métodos de desambiguação para outras áreas do PLN, sem aumentar demasiadamente o tempo de computação, algumas heurísticas podem ser aplicadas ([Agirre and Edmonds, 2006](#)). Dentre as abordagens heurísticas encontradas na literatura, [Mihalcea \(2006\)](#) apresenta o seguinte conjunto:

- um sentido por discurso: determinado o sentido de uma palavra P em um componente discursivo (sentença, parágrafo ou tópico do texto), todas as demais ocorrências de P possuirão o mesmo sentido;
- um sentido por colocação: determinado o sentido de uma palavra pelo seu uso, referente à forma na qual foi usada, é aplicado este sentido para as demais ocorrências deste mesmo uso (colocação) da palavra alvo. Por exemplo, ao atribuir o sentido de “instituição financeira” à palavra “banco”, precedida pelo pronome “ao”, todas essas ocorrências posteriores terão o mesmo significado; e
- sentido mais frequente: neste método todas as palavras são desambiguadas com seus sentidos mais frequentes em um *córpus* anotado ou em um repositório de sentidos.

Ressalta-se que essas heurísticas também podem ser estendidas para o contexto multi-documento. Por exemplo, dado uma coleção de N textos e uma palavra P muito frequente nesta coleção, o método de DLS poderia determinar o sentido de P em K textos, onde $K < N$, e, para os demais textos subsequentes, atribuir à palavra P o seu sentido mais frequente nos K textos anteriores. Essa heurística pode ser bastante útil para desambiguar grandes coleções de textos relacionados.

Os métodos de DLS podem, ainda, ser classificados entre abordagens baseadas em conhecimento, baseadas em *córpus* e abordagens híbridas. Abordagens baseadas em conhecimento utilizam recursos linguísticos como fontes de conhecimento para a desambiguação e determinam o sentido correto por meio de algoritmos que, geralmente, empregam, de alguma forma, métricas de similaridade. A abordagem baseada em *córpus* caracteriza-se pela criação de classificadores por meio da utilização de AM em *córpus* anotados. Por fim, abordagens híbridas agrupam características das duas abordagens supracitadas (Agirre and Edmonds, 2006).

Geralmente, métodos baseados em conhecimento são mais abrangentes, ou seja, são capazes de desambiguar uma quantidade maior de palavras e, normalmente, são mais facilmente portáteis para outras línguas, alterando-se os recursos utilizados. Métodos baseados em *córpus* requisitam tempo maior de desenvolvimento, visto a necessidade de treinamento de técnicas de AM e, muitas vezes, compilação de *córpus* anotados. Estes métodos são mais restritos, o que se justifica por: ser necessária a criação de um classificador para cada palavra; somente os sentidos presentes no conjunto de testes podem ser adotados pelo classificador, ou seja, se uma palavra possui 5 possíveis sentidos, porém, no conjunto de testes há apenas 2, os classificadores são capazes de optar por apenas um destes.

Apesar dos métodos baseados em conhecimento serem mais abrangentes, normalmente, apresentam resultados inferiores aos alcançados pelos métodos baseados em *córpus*. Con-

tudo, devido a esta característica de maior abrangência e no intuito de atingir os objetivos deste trabalho, desenvolver métodos de DLS de tal forma que estes possam ser empregados em outras aplicações do PLN e em diversos contextos, optou-se por métodos baseados em conhecimento.

Na próxima seção, serão apresentadas as formas de avaliação para métodos de DLS. Essa tarefa é apontada por alguns autores como difícil de ser executada no âmbito da DLS, devido à diversidade presente nos sistemas de DLS, como: tarefa destinada (*lexical sample* ou *all words*), idioma, aplicação final do método de DLS (genérico, Tradução Automática, Recuperação de Informação, Sumarização Automática, etc.). Também serão apresentadas iniciativas de padronização de avaliações, como métricas estipuladas e eventos como o Semeval¹.

2.3 Avaliação de Sistemas em DLS

A avaliação de métodos de DLS é uma etapa bastante importante, sendo apontada por Edmonds (2002) como etapa crucial para o êxito dos algoritmos desenvolvidos. Contudo, trata-se de um processo complexo, principalmente quando se deseja comparar dois métodos, visto as características diferentes entre estes, por exemplo: quais palavras são desambiguadas pelo sistema, qual o RS utilizado, qual tarefa o sistema aborda (amostra de palavras, todas as palavras, etc.), etc. (Jurafsky and Martin, 2009). Outro ponto importante é que, muitas vezes, a DLS é empregada em etapas de pré-processamento para outras áreas de pesquisa.

Nesse contexto, tem-se duas formas de avaliação, extrínseca e intrínseca. Na avaliação extrínseca ou validação, o objetivo é avaliar a contribuição do emprego de métodos de DLS em outras aplicações. Jurafsky and Martin (2009) relatam que tal abordagem de avaliação é complexa e necessita de maior tempo, devido à necessidade de metodologias de experimentação adequadas para avaliar o desempenho da DLS em outras áreas de pesquisa. Já na segunda forma de avaliação, a intrínseca, os métodos de DLS são testados de forma independente, ou seja, deseja-se avaliar a qualidade de desambiguação do método proposto.

Geralmente, são utilizados corpúscos anotados ou conjuntos de testes para subsidiar a avaliação, seja pela metodologia extrínseca ou intrínseca. Assim, o processo de avaliação ocorre em duas fases: (1) aplica-se o método de desambiguação no conjunto de testes; e (2) verifica-se a correspondência entre os sentidos presentes no conjunto de testes e os sentidos indicados pelo método de DLS por meio de alguma métrica de avaliação. Além disso, geralmente, no processo de validação, é necessário executar mais etapas de testes

¹<http://semeval2.fbk.eu/semeval2.php>.

para avaliar as consequências do uso da DLS em outras aplicações.

As métricas utilizadas para avaliar métodos de DLS são enumeradas a seguir, sendo que: npC é o número de palavras corretamente desambiguadas pelo sistema; npD é o número de palavras que tiveram algum sentido atribuído, correto ou não; npT é o número de palavras do conjunto de teste que deveriam ser desambiguadas; e npC' representa o número de palavras corretamente desambiguadas pelo método com auxílio do sentido majoritário, ou seja, caso o método não determine o sentido de uma palavra do conjunto de testes, adota-se o sentido mais frequente para esta.

- precisão: percentual de palavras corretamente desambiguadas em relação à quantidade de palavras do conjunto de teste às quais foi atribuída alguma classificação pelo sistema:

$$\frac{npC}{npD} \quad (2.1)$$

- cobertura: percentual de palavras corretamente classificadas pelo método em relação a todas as palavras do conjunto de teste que deveriam ser desambiguadas:

$$\frac{npC}{npT} \quad (2.2)$$

- abrangência: percentual de palavras que foram desambiguadas, independente da desambiguação ser correta ou não:

$$\frac{npD}{npT} \quad (2.3)$$

- acurácia: segundo [Specia \(2007\)](#), essa métrica é calculada por meio do percentual de palavras corretamente desambiguadas pelo sistema com relação ao total de palavras, usando o sentido majoritário (mais frequente) quando o sistema não consegue desambiguar alguma palavra:

$$\frac{npC'}{npT} \quad (2.4)$$

Geralmente, são utilizadas medidas de referência, denominadas *baselines*, para avaliar métodos de DLS. Essas medidas representam, usualmente, um limite inferior de qualidade para o método. Para calcular a *baseline*, aplicam-se métodos mais simples de DLS em um conjunto de testes e aferem-se os resultados conforme as métricas anteriores. Métodos frequentes na literatura para calcular estas medidas são: atribuir o primeiro sentido de uma *wordnet*, atribuir à palavra ambígua o sentido mais frequente em um *corpus*, e aplicar o algoritmo de [Lesk \(1986\)](#).

Devido à diversidade de características inerentes em algoritmos de DLS, a prática da avaliação comparativa perdeu forças na área ([Mitkov, 2004](#)). Contudo, deve-se ressaltar

que comparações podem ser realizadas, desde que parâmetros relacionados a recursos e objetivos dos sistemas sejam considerados. Uma iniciativa de se padronizar a avaliação de sistemas em DLS vem sendo aplicada pelo SemEval.

O SemEval constitui-se em um evento criado para avaliar sistemas em tarefas que envolvam DLS (Mihalcea and Edmonds, 2011). No ano de 2012, ocorreu a sexta edição do evento, o SemEval-3². O evento assemelha-se a uma competição, devido à especificação de tarefas, que representam contextos e objetivos para métodos de DLS. Em geral, essas tarefas são constituídas por: conjunto de dados para testes, conjunto de dados para treinamento (para sistemas de AM), formas de avaliação, objetivos (qual tarefa, quais palavras devem ser desambiguadas, qual idioma, etc.), recursos disponíveis e valores de *baseline*.

As edições mais recentes do evento abordaram temas mais diversos, mas que possuem alguma relação com a área de DLS. Por exemplo, na edição de 2013 (semeval-2013)³, as tarefas propostas incluem: análise de sentimentos em tweets; similaridade de semântica textual; dentre outras tarefas. Essas tarefas envolvem a DLS de forma implícita, o que afirma, de certa forma, que a DLS está presente em diversas aplicações.

2.4 Recursos e Ferramentas Utilizadas em DLS

Nesta Seção, são apresentados os principais recursos e ferramentas utilizados em sistemas de DLS. Os recursos apresentados na Seção 2.4.1, em geral, são aplicados como fontes de conhecimento linguístico durante o processo de desambiguação ou para treinamento de métodos de AM. Na Seção 2.4.2, são apresentadas as principais ferramentas utilizadas, que normalmente são empregadas para pré-processar os objetos de entrada (sentença, texto, etc.) dos sistemas de DLS.

2.4.1 Recursos

Nesta seção, são apresentados os principais recursos utilizados em sistemas de DLS. Esses recursos são empregados como fontes de conhecimento linguístico ou repositório de sentidos, seja em fase de desenvolvimento, avaliação ou durante a desambiguação. Os recursos são apresentados nesta ordem: *corpus*, dicionários, *thesaurus* e *wordnets*.

Corpus, segundo Sardinha (2005), consiste em uma coletânea de dados linguísticos naturais (escritos por seres humanos) organizados para serem legíveis por computador. Entretanto, existem diversas outras definições na literatura. Por exemplo, define-se *corpus* como uma coleção de objetos linguísticos para uso em aplicações de PLN (Santos,

²<http://www.cs.york.ac.uk/semeval-2012/>

³<http://www.cs.york.ac.uk/semeval-2013/>

2008). Os objetos armazenados, geralmente textos, são selecionados conforme um domínio específico para atender algum cenário de aplicação. Sua organização deve permitir que sistemas, ou usuários humanos, façam consultas para extrair algum tipo de conhecimento, por exemplo, palavras ou expressões mais frequentes.

Outra característica interessante do trabalho com *corpus* é a possibilidade de anotação, que consistiu-se por uma tarefa (geralmente, manual) para explicitar diferentes níveis de conhecimento ao rotular os objetos armazenados no *corpus*. O tipo de informação inserida e quais os componentes (palavras, sentenças, textos, etc.) que serão anotados variam conforme o objetivo de uso do *corpus*. Em específico para aplicações de DLS, as palavras são anotadas com seus respectivos sentidos.

Pesquisas em DLS utilizam *corpus* anotados tanto para treinamento (de métodos baseados em Aprendizado de Máquina) quanto para avaliação dos métodos. Na avaliação, os exemplos (palavras com respectivos sentidos e contextos) presentes no *corpus* são anotados automaticamente por algum método de DLS e, posteriormente, verificam-se as correspondências entre os sentidos atribuídos pelo método e os anotados no *corpus* por meio de alguma métrica de avaliação. Já em fase de desenvolvimento, os *corpus* são tratados, geralmente, como um conjunto de exemplos rotulados para treinamento de técnicas de AM. Dentre os recursos disponíveis na literatura, pode-se citar o SemCor⁴ (Mihalcea, 1998) e o OntoNotes⁵ (Hovy et al., 2006), que serão apresentados a seguir.

O SemCor foi criado na *Princeton University*, é constituído por 360 mil palavras, originadas de 352 textos que são um subconjunto do *corpus* Brown. O SemCor possui anotações da classe gramatical, lema, e *synsets* da Wn-Pr. Atualmente, encontra-se na versão 3.0, com mais de 200 mil palavras anotadas com os *synsets* da Wn-Pr (Melli's, 2011) .

O OntoNotes é um projeto de colaboração entre a empresa *Raytheon BBN Technologies*, a *University of Colorado*, a *University of Pennsylvania* e o *University of Southern California's Information Sciences Institute*, cujo objetivo foi a criação de um grande *corpus* semanticamente anotado em diversos idiomas. O *corpus* abrange vários gêneros textuais (notícias, conversas de telefone, blogs da Web, transmissões de jornais, programas de TV) em três idiomas: Inglês, Chinês e Árabe. Atualmente, encontra-se em sua quarta versão (OntoNotes Release 4.0). O *corpus* é anotado com *synsets* da Wn-Pr, além de possuir também anotação da classe gramatical das palavras (Weischedel et al., 2010).

Neste trabalho, será utilizado o *corpus* CSTNews (Aleixo and Pardo, 2008; Cardoso et al., 2011), composto por textos jornalísticos organizados em 50 coleções de documentos, com dois ou três textos. O *corpus* possui 72.148 palavras dispostas em 140 documentos

⁴Disponível em: http://www.gabormelli.com/RKB/SemCor_Corpus

⁵Disponível em: <http://www.bbn.com/ontonotes/>

agrupados conforme o tema abordado. O CSTNews é anotado com etiquetas referentes a: duas teorias discursivas, a *Rhetorical Structure Theory* (RST) (Mann and Thompson, 1987) e a *Cross-document Structure Theory* (CST) (Radev, 2000); segmentação topical (Cardoso et al., 2012); sumários mono e multidocumento (Aleixo and Pardo, 2008); alinhamento sentencial entre textos-fontes e sumários (Agostini et al., 2012). Porém, não possui anotações dos sentidos das palavras. Portanto, para sua aplicação em métodos de DLS, torna-se necessária uma etapa de anotação, para adicionar o sentido das palavras ao corpus.

A utilização desse corpus dá-se pelas seguintes razões: 1) é um recurso desenvolvido pelo grupo de pesquisa NILC, do qual o autor deste trabalho é integrante; 2) a anotação semântica, desambiguando as palavras do corpus por meio dos *synsets* da Wn-Pr, pode contribuir para futuras pesquisas para o Português do Brasil, principalmente no NILC; 3) os textos jornalísticos, que compõem o corpus, são escritos em uma linguagem mais comum e difundida; e 4) apresenta coleções de textos sobre um mesmo tópico, o que contribui para a investigação e desenvolvimento de métodos de DLS no contexto multidocumento. Portanto, acredita-se que a utilização desse corpus, além de atender os objetivos deste trabalho, contribui para o desenvolvimento de métodos de desambiguação no cenário multidocumento que, posteriormente, possam atender diversas outras aplicações do PLN.

É importante ressaltar que documentos da Web também podem ser utilizados como corpus. A grande quantidade de conteúdo diariamente publicado na Web permite a extração de algumas informações, principalmente quantitativas, que podem auxiliar algumas aplicações. Destaca-se, nesta abordagem, o método de Mihalcea and Moldovan (1999), que ranqueia os possíveis sentidos de uma palavra conforme a quantidade de páginas retornadas por um buscador Web.

Dicionários compreendem um conjunto de palavras e suas respectivas definições. É importante ressaltar que, para aplicações do PLN, o uso de dicionários deve ser realizado por computador, portanto, devem-se utilizar dicionários que permitam tal interação, seja por meio de arquivos ou *software* de busca. Esses dicionários podem ser classificados como legíveis por máquina e manipuláveis por máquina. No primeiro caso, os dicionários são recursos computacionais que representam dicionários normais em formato digital, que não permitem que programas de computador acessem seus dados. Já no segundo caso, dicionários manipuláveis por máquina, permitem que programas de computador acessem os dados do dicionário.

Esse tipo de recurso pode ser bastante útil, pois disponibiliza definições e, em alguns casos, exemplos do uso de uma palavra. Essas informações podem ser utilizadas como contextos para auxiliar a desambiguação, como utilizado por Lesk (1986). Porém, Miller (1995) aponta que dicionários, mesmo os destinados a computadores, são especificados

para uma leitura humana e não para máquinas, conseqüentemente, proporcionam menos efetividade à DLS.

Thesauri, ou Tesouros, são espécies de dicionários, cujas palavras são organizadas conforme seus sentidos (Sardinha, 2005), isto é, as entradas são organizadas de tal forma que palavras que possuam um mesmo significado estejam em um mesmo conjunto. Normalmente, estes recursos possuem estruturas para organizar as palavras pela classe gramatical (verbos, substantivos, adjetivos, etc.) e também relacioná-las com respectivos antônimos. Assim, nestes repositórios, torna-se possível a busca de palavras semanticamente relacionadas, seja por sinonímia (compartilham um mesmo significado) ou antonímia (com sentidos contrários).

Um exemplo de *thesaurus* para o PT-BR é o Tep2.0⁶ (Maziero et al., 2008), composto por 44.678 itens lexicais distribuídos em 19.888 conjuntos de sinônimos e com 4.276 relações de antonímia. Para exemplificar a utilização de *thesaurus*, tem-se a utilização da palavra “calor” no TeP. O resultado da busca ⁷ é apresentando a seguir, sendo os antônimos, quando existentes, representados em subitens.

1. calor, aconchego, agasalho, conchego, conforto

2. calor, animação, aquecimento, entusiasmo

Antônimos: desanimação, desentusiasmo

3. calor, ardor, energia, fervor

Antônimos: mornidão, tepidez, tepor, tibiez, tibieza

4. calor, ardor, emoção, fervor, paixão

5. calor, acu, ardor, ardume, queimor, quentura

Antônimos: algidez, frialdade, frieza, frigidez, friúra, gelidez

Cada enumeração anterior representa um possível sentido para a palavra “calor”, que pode ser interpretado pelo conjunto de palavras sinônimas. Por exemplo, é perceptível que os sentidos 2 e 3 são diferentes, inclusive em seus antônimos. O item 2 representa um sentido mais abstrato da utilização da palavra “calor” do que o item 3, que representa um sentido mais físico.

As *wordnets* são bases de dados lexicais desenvolvidas especificamente para serem manipuladas por programas de computador. A representação de sentidos é dada com base em *synsets*, que são conjuntos compostos por palavras sinônimas, semelhante à estruturação dos tesouros. As palavras são agrupadas levando em consideração o significado e

⁶<http://www.nilc.icmc.usp.br/tep2/index.htm>

⁷Realizada por meio da interface web do Tep 2.0 em 20/02/2013 às 14:00.

classificação gramatical, ou seja, há *synsets* para substantivos, outros para verbos, etc. e uma palavra pode estar presente em ambos (Miller, 1995).

Wordnets são referenciadas por alguns autores como sendo Ontologias Linguísticas, pois são definidas por um conjunto de unidades lexicais, que representam conceitos, e suas relações (Di Felippo, 2008). Ressalta-se que a definição do termo ontologia é ambígua (Noy and McGuinness, 2001). Consequentemente, não há consenso entre a definição de *wordnets* como ontologias.

As relações entre *synsets* na *wordnet* ocorrem conforme a classe gramatical. Para substantivos são aplicadas basicamente quatro relações: hiperonímia, hiponímia, holonímia e meronímia (Banerjee and Pedersen, 2002). As duas primeiras representam relações de “tipo”; são relações que representam generalização e especificação, ou seja, quando um conceito C_1 é um tipo de C_2 (cadeira é um tipo de móvel, por exemplo), diz-se que C_1 é hipônimo de C_2 , e, por outro lado, diz-se que C_2 é hiperônimo de C_1 . As outras, holonímia e meronímia, representam relações “parte de”, ou seja, quando um conceito C_1 constitui um conceito C_2 (tijolos fazem parte de uma casa, por exemplo), diz-se que C_1 é merônimo de C_2 e que C_2 é holônimo de C_1 .

A WordNet de Princeton é compilada para o idioma Inglês Americano (Fellbaum, 1998). Para o Português do Brasil, tem-se a WordNet-BR⁸ em fase de desenvolvimento e sua especificação contempla um mapeamento entre seu *synsets* com a WordNet de Princeton (Dias da Silva, 2005).

A Onto.PT⁹ trata-se de um recurso semântico destinado à língua Portuguesa, considerada na literatura, assim como sugere seu título, um ontologia linguística, mas possui estrutura semelhante à de uma *wordnet*. Este RS foi compilado de forma semi-automática por meio de recursos como dicionários, tesouros e páginas Web. É composta por, aproximadamente, 150 mil itens lexicais, organizados em 110 mil *synsets* e com 178 mil relações entre estes (Oliveira and Gomes, 2012).

2.4.2 Ferramentas

Em geral, para a DLS, algumas ferramentas são utilizadas para pré-processar os textos, possibilitando melhor desempenho dos sistemas de desambiguação. Porém, algumas abordagens utilizam ferramentas disponíveis como componentes de seus sistemas.

Anotadores morfossintáticos, ou *taggers*, como normalmente são referenciados na literatura, são ferramentas computacionais desenvolvidas para rotular as palavras de uma sentença ou texto com as respectivas classes gramaticais. Por exemplo, na sentença “O

⁸<http://www.nilc.icmc.usp.br/wordnet/>

⁹Disponível em: <http://ontopt.dei.uc.pt/>

banco quebrou”, o *tagger* deveria classificar as palavras desta sentença em artigo, substantivo comum e verbo, respectivamente.

Neste trabalho, foi adotado o MXPOST, um *tagger* automático com algoritmo independente de língua, desde que haja treinamento, desenvolvido na linguagem de programação Java®. Seu funcionamento baseia-se em um modelo probabilístico pautado no cálculo da entropia (Ratnaparkhi, 1986). Em experimentos realizados por Ratnaparkhi (1986), a precisão deste *tagger* chega a 96.49% para o idioma Inglês. Para o Português do Brasil, experimentos realizados por Aires (2000), que utilizaram um corpus anotado de aproximadamente 100 mil palavras e um conjunto de 17 *tags* possíveis (incluindo *tags* de pontuações), apresentaram uma performance de 97% de acurácia.

Para utilização do MXPOST, é indispensável o treinamento de seu algoritmo. Para tanto, são necessários dois conjuntos: de treinamento; e de etiquetas (*tag set*). O conjunto de treinamento, normalmente um corpus, corresponde a um acervo de palavras previamente etiquetadas. O *tag set* corresponde a uma listagem das possíveis etiquetas para os itens lexicais, portanto, as etiquetas empregadas no conjunto de treinamento devem estar incluídas no *tag set*. Após a etapa de treinamento, além do texto ou sentença que se deseja etiquetar, deve também ser informado um *tag set* para utilização do MXPOST. Os recursos utilizados para viabilizar o uso do MXPOST neste trabalho estão disponíveis na página do projeto NILC’s Taggers¹⁰.

Para exemplificar a utilização do MXPOST, tem-se a seguinte sentença “O banco quebrou”. Após execução da ferramenta, é apresentada a seguinte saída: “O_ar banco_nn quebrou_v”. É importante informar que, por padrão, este etiquetador usa um símbolo de _ para separar uma palavra e sua respectiva etiqueta. Portanto, as sequências de etiquetas anotadas correspondem a: artigo, substantivo comum e verbo.

Tradutores automáticos consistem em ferramentas desenvolvidas para a tarefa de traduzir uma palavra em determinado idioma de origem para um idioma de destino. Por exemplo, traduzir palavras do Português para o Inglês. Neste trabalho, torna-se indispensável a utilização destas ferramentas para viabilizar buscas por *synsets* na Wn-Pr, pois as palavras a serem desambiguadas estão no idioma Português e os *synsets* são representados no idioma Inglês. Neste trabalho, foi adotado o dicionário bilíngue WordReference®¹¹.

O WordReference® foi selecionado porque, em geral, apresenta maior número de possibilidades de tradução. Com isto, espera-se que os *synsets* retornados abranjam todos os sentidos de uma palavra no PT-BR. Outras duas ferramentas de tradução foram cogitadas, o Google Translate®¹² e o Microsoft Translator®¹³. Porém, foram descartadas.

¹⁰<http://www.nilc.icmc.usp.br/nilc/tools/nilctaggers.html>

¹¹<http://www.wordreference.com/>

¹²<http://translate.google.com/>

¹³<http://www.microsofttranslator.com/>

O Google Translate®¹⁴, talvez, a ferramenta de tradução mais difundida, não foi adotada pois sua API de programação, que permite o desenvolvimento de programas para utilizar suas funcionalidades, é um recurso pago. A ferramenta Microsoft Translator® foi descartada após algumas avaliações manuais. Esta ferramenta, normalmente, apresenta apenas a tradução mais frequente da palavra, o que prejudica a listagem dos possíveis *syn-sets* para uma palavra. Por exemplo, para a palavra “avião”, lista-se apenas a tradução “*plane*”. Já no WordReference®, são listadas as possibilidades “*plane*” e “*airplane*”.

Ferramentas de busca são sistemas, normalmente em plataforma Web, que, dada uma chave de busca, retornam uma listagem de documentos relevantes. Este tipo de ferramenta é empregado em alguns métodos de DLS, como o método de Mihalcea and Moldovan (1999), que foi adaptado neste trabalho, para desambiguar as palavras.

Neste trabalho, foi utilizado Bing®¹⁴. Outras ferramentas mais conhecidas, como o Google Search®¹⁵, não foram adotadas pois, assim como o Google Translate®, possuem API de desenvolvimento paga.

¹⁴<http://br.bing.com/>

¹⁵<https://www.google.com.br/>

Trabalhos Relacionados

Neste capítulo será apresentada uma revisão de trabalhos relacionados à DLS. Os métodos de DLS são agrupados conforme suas abordagens em duas seções, abordagens baseadas em conhecimento (Seção 3.1) e abordagens baseadas em *cópus* (Seção 3.2).

Métodos de DLS podem ser aplicados para desambiguar sentenças, textos, palavras isoladas, etc. Portanto, no decorrer deste capítulo, será aplicado o termo “entrada” para referenciar textos ou sentenças. Outro termo que, frequentemente, será aplicado no texto é “palavra alvo”, que corresponde à palavra que se deseja desambiguar.

3.1 Abordagens Baseadas em Conhecimento

Lesk (1986) apresenta um método de DLS que faz uso de dicionário como repositório de sentidos. O método proposto, conhecido como algoritmo de Lesk, parte do princípio que o sentido adequado para uma palavra é aquele que possui maior similaridade com o contexto. Para tanto, computa-se, para cada possível sentido, o número de sobreposições de palavras entre as respectivas definições dos sentidos (extraídas de dicionários) e rótulos atribuídos às palavras do contexto. Assim, o significado com maior número de sobreposições é adotado.

Os rótulos são *bag-of-words* constituídas por palavras extraídas das definições (glosa, por exemplo) de todos os possíveis sentidos de uma palavra. Por exemplo, uma palavra p possui n entradas no dicionário, portanto, seu rótulo é constituído pelas palavras que constituem as glosas de seus n sentidos no dicionário.

O algoritmo de Lesk se caracteriza como um método independente de língua e de

grande abrangência, pois é capaz de desambiguar qualquer palavra que esteja listada em algum dicionário e em qualquer idioma. Entretanto, a complexidade computacional para o cálculo das sobreposições de palavras pode ser um limitante para desambiguação de grandes textos. Por exemplo, para desambiguar uma palavra P por meio de uma janela com 10 palavras, onde cada palavra na entrada possui rótulos de tamanho n (quantidade de palavras que compõem o rótulo), são necessárias $10kn^2$ comparações de palavras. Isso ocorre pois, sendo que P possui k sentidos com n palavras cada, é necessário comparar cada um dos 10 rótulos com n palavras para cada definição de P .

O método de Lesk também não considera os sentidos das palavras anteriores. Ou seja, ao desambiguar uma palavra i , seu sentido correto não é empregado para desambiguar uma palavra j , com $i < j$. Utilizar os sentidos anteriores poderia diminuir a complexidade computacional do método, pois a quantidade de comparações necessárias seria menor. Outra limitação do método está na possibilidade de privilegiar algumas palavras. O cálculo da sobreposição de palavras não considera a extensão das definições utilizadas no dicionário. Assim, palavras que possuam definições mais extensas podem apresentar mais sobreposições em seu contexto, ou seja, privilegiadas pelo método (Specia, 2007).

Lesk não apresenta experimentos sistemáticos para seu método. Porém, alguns testes realizados pelo autor, que foram manualmente elaborados para o idioma Inglês, apresentam valores de acurácia entre 50% e 70%. Todavia, ressalta-se que este método é bastante citado na literatura sendo frequentemente empregado em avaliações comparativas com outros métodos de DLS.

Yarowsky (1995) investiga heurísticas de desambiguação a fim de impedir trabalho redundante (aplicação de algoritmos em contextos semelhantes) e, consequentemente, diminuir a complexidade computacional dos algoritmos. A hipótese parte do princípio que ocorrências de uma palavra presente em contextos significantemente semelhantes apresentam o mesmo sentido. Assim, propôs duas heurísticas: um sentido por discurso; e um sentido por colocação. Para a primeira abordagem, deve-se definir um componente discursivo (sentença, parágrafo, tópico textual, texto, etc.) e, após adotar um sentido s para uma palavra p em um componente C , todas as demais ocorrências de $p' \in C$ são desambiguadas por meio do sentido s . A segunda abordagem visa analisar a forma de utilização de uma palavra p e atribuir o mesmo significado às demais ocorrências que foram similarmente empregadas, por exemplo: uso de plural ou singular; sintagma verbal mais próximo; preposições antes ou depois; posição textual, etc.

Mihalcea and Moldovan (1999) apresentam uma abordagem de DLS por meio de buscadores Web. Seu método se propõe à desambiguação usando pares de palavras para construção de *queries* de busca. Entretanto, não deixam claro como os pares de palavras são extraídos da entrada. Segundo os autores, tal abordagem é viável devido à grande

quantidade de textos disponibilizados na Web, o que possibilita extrair informações sobre o uso de determinadas palavras em conjunto.

O funcionamento deste método, assim como o método de Lesk supracitado, está na hipótese de que: dentre os possíveis sentidos de uma palavra, o correto é aquele que possui maior similaridade com o contexto em que esta é empregada, sendo sua palavra par empregada como contexto e a medida de similaridade aferida pelo número de resultados retornados por uma ferramenta de busca. Ou seja, os autores utilizam a Web como *cópus* para verificar qual sentido mais co-ocorre com o contexto (palavra par) da palavra alvo.

Os autores apresentam dois algoritmos que se caracterizam por ranquear os possíveis significados (*synsets* de uma *wordnet*) das palavras, ou seja, os algoritmos após realizar buscas na Web apresentam quatro possíveis sentidos para uma palavra, sendo estes listados por ordem de relevância. O primeiro algoritmo pode ser descrito nas seguintes etapas:

1. dado um par de palavras ($P1, P2$), procuram-se todos os *synsets* da Wn-Pr que contenham $P2$;
2. para cada *synset* retornado, constroem-se pares formados pelos sinônimos de $P2$ e a palavra $P1$; e
3. para cada *synset*, constroem-se chaves de busca usando os respectivos pares de palavras obtidos na etapa 2. Posteriormente, realizam-se buscas na Web por meio de uma ferramenta de busca (os autores empregam o ALTAVISTA¹). A quantidade de resultados obtidos nas buscas ranqueia os possíveis sentidos. A construção das chaves de busca para cada *synset* segue dois formatos: (a) “ $P1 S^1$ ” OR “ $P1 S^2$ ” OR ... OR “ $P1 S^n$ ” e (b) ($P1$ NEAR S^1) OR ($P1$ NEAR S^2) OR ... OR ($P1$ NEAR S^n), onde S^i representa o i -ésimo sinônimo de $P2$ no respectivo *synset*.

Por exemplo, para desambiguar o par de palavras *broke* e *bank*, sendo um verbo e um substantivo, respectivamente, tem-se a seguinte utilização das etapas:

etapa 1: listam-se todos os *synsets* da palavra *bank* – para exemplo, apenas os dois primeiros *synsets* serão apresentados;

1. bank
2. depository financial institution, bank, banking concern, banking company
3. bank (a long ridge or pile)
4. bank (an arrangement of similar...

¹<http://www.altavista.com/>

5. bank (a supply or stock held
6. bank (the funds held by a gambling...
7. bank, cant, camber (a slope in the turn...
8. savings bank, coin bank, money box, bank
9. bank, bank building
10. bank (a flight maneuver;...

etapa 2: são formados os pares com a palavra *broke* e os sinônimos de *bank*. Portanto, os pares formados são: (*broke, bank*) para o primeiro *synset* e (*"broke, depository"*), (*"broke, financial institution"*), (*"broke, bank"*), (*"broke, banking concern"*) (*"broke, banking company"*) para o segundo *synset*; e

etapa 3: as chaves de buscas criadas usando o formato (a) seriam: (*"broke bank"*) e (*"broke depository" OR "broke financial institution" OR "broke bank" OR "broke banking concern" OR "broke banking company"*).

O algoritmo supracitado foi testado com 384 pares de palavras extraídos do SemCor 1.6. A relação desses pares de palavras quanto à classe gramatical foi de 200 pares de verbos e substantivos; 127 pares de adjetivos e substantivos e 57 pares de advérbios e verbos. Após experimentos que consideravam o ranqueamento atribuído aos *synsets*, a menor taxa de acerto alcançada foi de 60% para desambiguação de verbos quando se utiliza o sentido melhor ranqueado (que obteve mais resultados na busca). Já a melhor taxa de acerto foi de 98% para desambiguação de substantivos, contudo, tal valor de precisão foi alcançado por meio do *synset* presente na quarta posição do ranque.

O segundo algoritmo destina-se à desambiguação de pares de palavras formados por um verbo e um substantivo. Sua principal diferença para o primeiro algoritmo é o uso do conceito de Densidade Semântica. Esse conceito é uma métrica que representa a distância semântica entre duas ou mais palavras.

Para calcular a densidade semântica, os autores utilizam a Wn-Pr e aplicam a Equação 3.1, que calcula a sobreposição de palavras entre as glosas da palavra, assim como no método de Lesk (1986), levando em consideração o nível em que estas glosas estão na *wordnet*, ou seja, quanto mais longe um *synset* está de uma palavra, menor é o peso da quantidade de palavras sobrepostas. As variáveis que compõem a equação são: $|cd_{ij}|$, que é a quantidade de substantivos comuns que ocorrem simultaneamente na hierarquia do verbo e do substantivo, w_k , que representa o nível em que se encontram os substantivos na hierarquia do verbo e, $descendents_j$, que representa a quantidade de palavras encontradas na hierarquia do substantivo.

$$C_{ij} = \frac{\sum_k^{|cd_{ij}|} w_k}{\log(descendents_j)} \quad (3.1)$$

A hierarquia de verbos é definida por todos os *synsets* entre o verbo e seu mais alto hiperônimo, sendo incluídos também os hipônimos diretos. Já a hierarquia de substantivos é representada por todos os substantivos que estão na classe do substantivo a ser desambiguado. Os autores não deixam claro como realizar o processo para definir a hierarquia de substantivos. Entretanto, ressaltam que alguns substantivos possuem grandes hierarquias, e por isso empregam a função $\log(descendents_j)$ para normalizar os dados.

O segundo algoritmo definido pelos autores é representado nas seguintes etapas:

1. anotam-se os possíveis sentidos para verbo e substantivo por meio de *synsets* da Wn-Pr;
2. usa-se o algoritmo 1 para ranquear os sentidos do substantivo;
3. para cada possível par (de sentidos entre verbo e substantivo), computa-se a densidade semântica; e
4. ranqueiam-se os sentidos.

Os autores aplicam o segundo algoritmo para desambiguação de pares de palavras formados por um verbo e substantivo. Para desambiguação de verbos, o segundo algoritmo obteve melhora nos dois primeiros sentidos do ranque, com uma taxa de acerto de 67% e 79% contra 60% e 68% respectivamente para o primeiro algoritmo. Já para desambiguação de substantivos, o segundo algoritmo obteve melhora nos três sentidos melhor ranqueados, com 86.5%, 96% e 97% contra 76%, 83% e 86% respectivamente do primeiro algoritmo.

Os autores apresentam duas limitações de seus métodos. A primeira limitação está na falta de anotação morfossintática das glosas da Wn-Pr. Assim, para calcular a densidade semântica sobre as glosas, torna-se necessário usar um *tagger* automático, que pode gerar erros de anotação e, conseqüentemente, interferir no cálculo da densidade semântica e nos resultados do segundo algoritmo. A segunda limitação é a falta de contexto para as palavras, pois seu método se propõe à desambiguação somente entre pares de palavras, e o emprego desses pares em uma sentença ou parágrafo poderia indicar mais pistas do sentido correto. Entretanto, a abordagem proposta por [Mihalcea and Moldovan \(1999\)](#) demonstra-se eficiente e rápida (para desenvolvimento), pois não possui exigência de trabalho de anotação manual de corpus. E, como apontam os autores, seu método pode ser empregado para limitar os possíveis sentidos de uma palavra, conseqüentemente, contribuindo para diminuir a complexidade computacional de outros métodos.

Kilgariff et al. (2000), em relatório sobre os resultados e conclusões do Senseval-1, edição de 1998, apresentam uma versão simplificada do algoritmo de Lesk (1986), visando menor complexidade computacional. Esta simplificação foi empregada para aferir valores *baseline* da competição.

Na versão original do algoritmo de Lesk (1986), o sentido adotado para uma PA é aquele que possui maior número de palavras comuns em sua definição do dicionário com os rótulos atribuídos às palavras no contexto da PA. A versão simplificada empregada no Senseval-1 pautou-se, principalmente, no descarte dos rótulos das palavras no contexto. Assim, o sentido selecionado corresponde àquele que possui mais palavras comuns em sua definição com as palavras no contexto.

Após avaliação dos trabalhos submetidos ao evento, a variação simplificada do algoritmo de Lesk apresentou precisão superior a nove métodos, sendo dois destes utilizados também como *baseline*. Ao analisarem a acurácia dos métodos, a simplificação proposta apresentou-se superior a quatorze trabalhos submetidos ao Senseval-1.

Banerjee and Pedersen (2002) apresentam uma adaptação do algoritmo de Lesk (1986). Esta constitui-se na aplicação da Wn-Pr como RS, não mais de dicionários lidos por computador, utilizados na versão original. Essa abordagem foi também implementada no *framework* WordNet::SenseRelate (WNSR) ² (Patwardhan et al., 2005).

Assim como no método de Lesk, primeiramente, as palavras recebem rótulos constituídos por uma lista de palavras oriundas de seus possíveis sentidos, neste caso, *synsets* da Wn-Pr. Para tanto, empregam as palavras existentes nas glosas, exemplos e conjunto de sinônimos dos *synsets*. Entretanto, afirmam que, devido às distintas constituições destas informações nos *synsets*, a utilização de uma ou outra informação para criação dos rótulos pode privilegiar alguns *synsets*. Por exemplo, ao utilizar somente palavras oriundas dos exemplos, podem haver *synsets* com poucos ou nenhum exemplo, o que, consequentemente, elimina-os da possibilidade de serem adotados pelo método.

Após se rotular as palavras, para cada palavra alvo, computa-se a similaridade de seus possíveis *synsets* com os rótulos das palavras ao redor. Esta similaridade é computada por meio do cálculo da quantidade de palavras iguais que ocorrem tanto nas informações do *synset* (glosas, exemplos e/ou palavras sinônimas) quanto nos rótulos no contexto. Posteriormente, o sentido com maior valor de similaridade é adotado.

Para realização de experimentos, os autores utilizaram o conjunto de testes do Senseval-2 destinado à tarefa de *lexical sample*. Esse conjunto de testes possui 73 palavras, sendo 29 substantivos, 29 verbos e 15 adjetivos. Ao final dos experimentos, o valor médio de acurácia obtido foi de 31.7%, onde 1374 palavras foram corretamente desambiguadas de 4328 casos de teste.

²<http://www.d.umn.edu/~tpederse/senserelate.html>.

Vasilescu et al. (2004) apresentam experimentos para avaliar variações na configuração do algoritmo de Lesk (1986). Neste intuito, verificam variações de tamanho de contexto, descrição de sentidos, pontuação com diferentes pesos e cadeias lexicais. Ainda neste cenário, aplicam o algoritmo *Naïve Bayes* para determinar as palavras com maior probabilidade de co-ocorrência em uma mesma janela, a fim de aplicar o algoritmo de Lesk nestes pares.

No primeiro atributo, utilizam contextos simétricos, ou seja, n palavras à esquerda e à direita da palavra alvo, sendo utilizados os valores 2, 3, 8, 10 e 25. Os autores ainda apontam que contextos simétricos, apesar de serem os mais comuns, não são tão eficientes para desambiguação de verbos, e que, normalmente, as palavras à direita possuem maior importância no processo de DLS. É importante ressaltar que esse atributo foi avaliado com os possíveis valores de n supracitados para todas as variações do método de Lesk apresentadas pelos autores.

Quanto à definição do sentido que é utilizada para construção dos rótulos (Lesk, 1986), os autores apresentam três variações: DEF, quando somente são utilizadas informações das glosas dos *synsets* da Wn-Pr; REL, utilização das informações de um *synset* s bem como todo os demais que possuem relações de hiperonímia com s ; e por fim, a variação DEFREL, que emprega as duas anteriores.

As pontuações para os sentidos foram aplicadas de duas formas, com e sem pesos. No primeiro caso, referenciado pelos autores como BOW, a utilização dá-se igualmente ao método proposto por Lesk (1986), no qual a pontuação é dada pela quantidade de sobreposições de palavras entre um sentido e o contexto da palavra alvo. Já a segunda, doravante WHG, levando em consideração a discrepância entre as informações dos sentidos (alguns possuem glosas maiores do que outros, por exemplo), utilizaram pesos por meio de multiplicação logarítmica para normalizar a quantidade de sobreposições.

Os autores também verificaram a utilização de Cadeias Lexicais (CL), itens lexicais que possuem relações de coesão em um texto, empregando como contexto de uma palavra alvo, somente aquelas palavras que estão presentes em sua CL. Por fim, aplicam o algoritmo *Naïve Bayes*, sem utilização do método de Lesk, para determinar o sentido mais adequado de uma palavra, empregando como conjunto de testes o *corpus* SEMCOR.

Em seus experimentos comparativos, os autores partiram da versão original do algoritmo de Lesk (OL, de *Original Lesk*), que aplica o cálculo de sobreposições de palavras em rótulos derivados das definições (glosas) e exemplos dos *synsets*. Bem como uma variação simplificada (SL, de *Simple Lesk*), que emprega o cálculo entre a glosa do *synset* e as palavras (sem rótulos) no contexto.

Para cada uma das versões anteriores, os autores adicionam, gradativamente, os atributos WHG e CL. Para ambos os algoritmos, OL e SL, os resultados apresentam-se

melhores por meio da utilização de todos os atributos (WHG e CL). O algoritmo SL mostra-se superior sobre o OL em todos possíveis tamanhos de janelas. Em relação ao *Naïve Bayes*, esta superioridade não ocorre apenas na janela de tamanho 25.

É importante também ressaltar que o aumento da janela tende a denegrir a qualidade do classificador. Tal tendência não ocorre somente ao utilizar o algoritmo de *Bayes* e o atributo CL, os quais se apresentam melhores em janelas maiores. Em outros experimentos, os autores afirmam que janelas ideais são entre 4 e 6 palavras.

[Sinha and Mihalcea \(2007\)](#) apresentam uma abordagem de DLS por meio de medidas de similaridades aferidas em grafos. Para tanto, fazem uso da hierarquia e relações de *synsets* da Wn-Pr para construção de grafos, denominados pelos autores como grafos de dependências, e, posteriormente, verificam quais sentidos são mais relevantes por meio de métricas de centralidade em grafo.

O algoritmo é composto por três etapas. A primeira corresponde à construção das dependências no grafo (será detalhada a seguir). Após a criação das dependências no grafo, são aplicados os algoritmos de centralidade, que visam determinar os vértices mais relevantes em uma rede de conexões. Por fim, verificam-se as pontuações aferidas na etapa anterior de cada possível *synset* e adota-se aquele que possuir maior valor de centralidade.

O grafo é construído de tal forma que todos os possíveis sentidos sejam representados por vértices e as dependências entre estes por meio de arestas. Assim, para cada palavra no contexto, criam-se vértices no grafo para cada possível *synset*. Após a criação dos vértices, arestas são criadas entre vértices relacionados à palavras em uma mesma janela.

As arestas do grafo são ponderadas com valores numéricos que representam o nível de similaridade (disponíveis no pacote *WordNet::Similarity* ([Patwardhan et al., 2003](#))) entre os sentidos (vértices), sempre calculado entre pares. Dentre as possíveis medidas de similaridade, os autores empregam: lch ([Leacock and Chodorow, 1998](#)); lesk ([Lesk, 1986](#)); wup ([Wu and Palmer, 1994](#)); res ([Resnik, 1995](#)); lin ([Lin, 1998](#)) e jnc ([Jiang and Conrath, 1997](#)). Em algumas situações, arestas podem assumir valores iguais ou inferiores a zero e, consequentemente, são retiradas do grafo.

Foram utilizados quatro algoritmos de centralidade de grafo, que utilizaram as seguintes medidas: (1) quantidade de arestas que incidem sobre um vértice; (2) grau de proximidade, que corresponde à soma de todos os menores caminhos de um vértice a todos os demais no grafo; (3) grau de intermediação, que corresponde à quantidade de vezes que um vértice está presente em um menor caminho entre outros vértices do grafo; e (4) PageRank ([Brin and Page, 1998](#)), que será apresentando com mais detalhes no decorrer desta seção.

Os experimentos realizados pelos autores utilizaram o conjunto de testes do SenSeval-2 e SenSeval-3. O primeiro experimento teve por objetivo desambiguar apenas verbos e

substantivos, utilizando as métricas de similaridade para construção de grafo e apenas o algoritmo 1 de centralidade em grafo (arestas incidentes em um vértice).

Nesses primeiros experimentos, os autores concluem que a métrica de jcn apresenta-se melhor para desambiguação de substantivos, alcançando 85% de precisão. Ainda para esta classe morfossintática, a similaridade de lesk apresenta-se em segundo lugar, com 83% de precisão. Para verbos, a melhor métrica foi a lch com 66

Após este experimento, os autores avaliam a combinação das métricas de similaridade para todas as classes morfossintáticas, utilizando os melhores resultados no primeiro experimento. Para tanto, utilizam a jcn para arestas entre *synsets* de substantivos, a lch para sentidos de verbos e a métricas de lesk para as demais, incluindo entre classes distintas. Neste experimento, alcançaram 53,43

Para verificar o peso de cada métrica supracitada, realizam experimentos com jcn, lch, lesk e a combinação destas sem as regras de restrição de classes morfossintática. Com isto, relatam que o melhor método foi a utilização da métrica de lesk, com 51,87

Após elencarem as melhores métricas de similaridade com os experimentos, os autores passam a avaliar os diferentes algoritmos de centralidade (nos experimentos anteriores, apenas o algoritmo que emprega o grau de incidência no vértice foi utilizado). Utilizando o método anterior com as métricas de similaridades combinadas, obtiveram que os algoritmos de centralidade baseados no grau de incidência e PageRank são bem melhores que os demais, apresentando respectivamente 53,43% e 52,82% de precisão.

Com esta discrepância nos resultados, os autores adaptam um sistema de votação. Ou seja, usando os diferentes algoritmos de centralidade, o sentido para uma determinada palavra é aquele mais adotado por todos os métodos. Por exemplo, dada uma palavra com dois *synsets*, sendo o primeiro adotado por três métodos e o segundo apenas por um, adota-se o primeiro sentido. Com isto, a precisão geral aumenta para 54,86% (não sendo um valor tão significativo).

Por fim, os autores combinam seus melhores resultados anteriores, adotando o sistema de votação por meio dos algoritmos de centralidade e criando os grafos de dependências usando as métricas de similaridade com as regras de classes morfossintáticas (jcn para substantivos, lch para verbos, e lesk para as demais). Ao final do experimento, relatam que este método apresentou-se melhor, com 58,83% de precisão.

[Agirre and Soroa \(2009\)](#) utilizam o algoritmo PageRank ([Brin and Page, 1998](#)) em estruturas de grafos geradas a partir de *wordnets* e o texto de entrada de seu sistema, semelhante à abordagem proposta por [Sinha and Mihalcea \(2007\)](#). O algoritmo PageRank trata-se de um método para ranquear os vértices de um grafo de acordo com a sua estrutura de arestas. Assim, a utilização deste algoritmo no âmbito da DLS consiste em ranquear os possíveis sentidos para determinada palavra, usando a relação entre os possíveis sentidos

existentes em seu contexto (texto ou sentença). Esta abordagem pauta-se na hipótese que o algoritmo PageRank irá apontar os sentidos mais relevantes dentro de um contexto.

O ranque gerado pelo algoritmo consiste da soma de dois fatores: a quantidade de votos de um vértice e a probabilidade de alcançá-lo ao percorrer a estrutura do grafo. Os votos são atribuídos por meio das arestas, ou seja, se existe uma aresta e_{ij} entre os vértices v_i e v_j , um voto é atribuído a v_j levando em consideração o grau de importância (ranque) de v_i . O valor da probabilidade de se alcançar um vértice é calculado por uma matriz M , construída a partir de um grafo G com N vértices. A matriz M é constituída por N linhas e N colunas, onde cada célula M_{ji} é preenchida com o valor $\frac{1}{d_i}$ ou 0, caso exista uma aresta entre os vértices v_i e v_j ou não, respectivamente. O valor d_i representa a quantidade de arestas que saem do vértice v_i (assim, $\frac{1}{d_i}$ representa a probabilidade, sem adotar relevância, de sair de v_i e chegar diretamente a v_j).

Dadas as definições supracitadas, o valor do PageRank para cada vértice pode ser calculado por meio da Equação 3.2, onde v representa um vetor contendo todas as probabilidades de se alcançar um vértice, a variável MPr modela a quantidade de votos recebidos pelo vértice e c um valor entre 0 e 1 denominado fator de relaxamento (tradução da expressão *damping factor*).

$$Pr = cMPr + (1 - c)v \quad (3.2)$$

Na versão original do PageRank, todos os valores de v são $\frac{1}{N}$, assumindo igual probabilidade para se alcançar qualquer vértice em G . A versão modificada desse algoritmo, adotada por Agirre and Soroa (2009), assume probabilidades diferentes para o vetor v , especificando-as conforme os possíveis sentidos do texto de entrada.

Visto a utilização do algoritmo PageRank, o sistema apresentado pelos autores possui uma etapa de pré-processamento para construção do grafo. O grafo construído parte inicialmente da estrutura da Wn-Pr, definida como um grafo $G = (V, E)$, onde V é o conjunto de vértices que representam os sentidos (*synsets*) e E o conjunto de arestas não direcionadas que representam a relação $e_{i,j}$ entre dois sentidos v_i e $v_j \in V$. A entrada do algoritmo é uma lista de palavras W , onde cada palavra $w^i \in W$ relaciona-se com um conjunto de sentidos $S_i \subset V$, ou seja, cada palavra de entrada possui arestas para seus respectivos *synsets*.

Dado o grafo supracitado, nomeado pelos autores como G_{KB} (Grafo da base de conhecimento), os autores apresentam três maneiras de usar o algoritmo PageRank, uma com a versão original e duas com versões modificadas. No primeiro método, o algoritmo original – que assume igual probabilidade de se alcançar qualquer vértice no grafo – é aplicado em um subgrafo G_D (Grafo de desambiguação) de G_{KB} . Para criação de G_D , faz-se uma busca em profundidade em G_{KB} para cada sentido v_i associado a uma pala-

vra w_i , sendo que cada execução visa encontrar a distância mínima³ de um sentido v_i e os demais sentidos presentes no contexto de w_i . Assim, G_D é constituído pela união de vértices e arestas que pertencem aos caminhos mínimos de G_{KB} , portanto, se um sentido v_i possui 2 caminhos para v_j com distâncias de 2 e 5 arestas respectivamente, no grafo G_D o caminho mais longo será excluído.

Após a criação do grafo G_D , o algoritmo PageRank é aplicado. Ao fim de sua execução, os sentidos melhor ranqueados são atribuídos às palavras na lista de entrada. Em caso de empate entre dois ou mais sentidos, ambos são selecionados ou um dentre eles é escolhido aleatoriamente, conforme necessidades da aplicação.

A utilização de versões personalizadas do algoritmo PageRank ocorre na estrutura completa do grafo, oposto ao método anterior, que usa apenas um subgrafo G_D . Outra diferença no grafo está na criação das arestas entre as palavras do contexto e respectivos *synsets*, que passam a ser direcionadas⁴ no sentido palavra $\rightarrow$ *synset*.

A variação ocorre pela forma de preenchimento do vetor v (vide Equação 3.2), que representa as probabilidades de se alcançar um determinado vértice no grafo. Na versão original do algoritmo, todos os vértices possuem igual probabilidade. Na primeira variação, o Ppr, a probabilidade é concentrada nas palavras do contexto. Assim, as arestas dirigidas fazem com que as palavras mais importantes no contexto insiram mais votos nos *synsets*. Com isso, o algoritmo é aplicado para ranquear os sentidos.

Os autores relatam, após realizar alguns experimentos, que palavras com dois ou mais sentidos podem gerar erros na desambiguação, pois alguns sentidos mais relacionados se fortalecem e acabam tirando votos dos outros *synsets*. Com isso, os autores apresentam a segunda variação, o Ppr_w2w.

Nessa segunda variação do algoritmo, constroem-se grafos independentes para cada palavra no contexto e a probabilidade é inserida nas palavras em volta da palavra alvo. Isso faz com que os sentidos da palavra alvo sejam votados somente pelas relações das palavras em sua volta. Esse algoritmo resolve o problema dos sentidos que se reforçam mutuamente, porém, é capaz de desambiguar apenas uma palavra a cada execução.

Para avaliar seus métodos, os autores utilizaram os conjuntos de teste do Semseval-2 e Senseval-3. Ao utilizar o conjunto de teste do Senseval-2, o algoritmo Ppr_w2w obteve melhores resultados na maioria dos casos. Sendo a versão Ppr melhor para desambiguação de substantivos. Já para o conjunto de testes do Senseval-3, o algoritmo Ppr_w2w obteve melhores resultados para substantivos e adjetivos. A versão original do algoritmo PageRank apresentou melhor Precisão para desambiguação de verbos.

³Essa distância é medida pelo número de arestas necessárias para encontrar um vértice v_j a partir de v_i .

⁴Uma aresta direcionada, ou dirigida, e_{ij} , representa um caminho direto de v_i para v_j , mas não o contrário.

[Machado et al. \(2011\)](#) apresentam um método de desambiguação geográfica, ou seja, o objetivo do método é desambiguar palavras (ou n-gramas) que referenciam regiões, por exemplo, “São Paulo” que pode referenciar uma cidade, um estado, um time de futebol e um santo do catolicismo. Esse método de desambiguação serve para a melhoria de sistemas de Recuperação de Informação. Ressalta-se que esse método abrange diversos idiomas, inclusive o Português do Brasil.

Os autores apresentam uma ontologia, denominada *OntoGazetteer*, como recurso para desambiguação, que corresponde à representação de conceitos de regiões correspondentes a uma base de dados composta por: 150 mil lugares; 75 mil nomes alternativos; 200 mil relações entre estes lugares; e 247 termos relacionados. O processo de desambiguação das palavras de entrada corresponde a indexá-las aos conceitos relacionados na ontologia e, posteriormente, aplicar heurísticas para indicar o sentido correto.

Os experimentos apresentados utilizaram notícias jornalísticas, pois, segundo os autores, normalmente estas notícias fazem referências a duas ou mais regiões. Foram utilizados 160 documentos jornalísticos da cidade de Belo Horizonte, sendo que 152 deles apresentaram ambiguidade. Após realizar experimentos, os autores apontam que 81,9% das palavras foram corretamente desambiguadas.

[Nguyen and Ock \(2011\)](#) abordam a DLS por meio de algoritmos de otimização, utilizando a hierarquia de *synsets* da Wn-Pr como um grafo e adotando os sentidos mais adequados por meio do algoritmo de colônia de formigas. Para tanto, transformam o problema de desambiguação em um problema de caixeiro viajante.

O Caixeiro Viajante (CV) é um dos problemas clássicos da computação, que pode ser facilmente explicado fazendo uma analogia com um vendedor que deve cobrir algumas cidades. Assim, dado um grafo com vértices e arestas que, respectivamente, representem as cidades e as estradas (com devidos custos, ex.: distância) entre essas, o problema está em encontrar a melhor rota (menor custo de distância, por exemplo), partindo de uma cidade qualquer, que passe por todas as cidades indicadas apenas uma vez.

Para resolver a DLS por meio da solução do CV, os autores, primeiramente, criam um grafo de desambiguação, cujos vértices representam os possíveis *synsets* da Wn-Pr e as arestas representam a similaridade entre estes. Para cada palavra no contexto, são adicionados vértices no grafo para todos os possíveis *synsets*. Posteriormente, para cada par de vértices (*synset*) no grafo, são criadas arestas por meio de uma métrica de similaridade (utilizam o método de *lesk* ([Lesk, 1986](#))).

Assim como apresentado por [Banerjee and Pedersen \(2002\)](#), devido às distintas características dos *synsets* da Wn-Pr, os autores também empregam diversas informações para cálculo de similaridade, sendo estas: (glosa) as respectivas glosas dos *synsets*; (glosa + Wn) a glosa de um *synset* *s* adicionando as glosas de todos os *synsets* relacionados a

s ; (glosa + XWN) assim como o item anterior, porém utilizam a extensão da WordNet (Mihalcea and Moldovan, 2001); (glosa + wiki) a glosa com mais informações de artigos associados na Wikipédia; e a união de todos.

Após a criação do grafo de desambiguação, emprega-se o algoritmo de colônia de formigas para resolução do CV. Ou seja, dado o conjunto de vértices que representam os possíveis *synsets* e as arestas que os relacionam, o algoritmo objetiva encontrar uma rota que passe somente uma vez por cada palavra no contexto. Assim, tendo em vista que uma palavra é visitada quando um de seus possíveis *synsets* é visitado, ao final do algoritmo, tem-se uma rota com apenas um *synset* para cada palavra no contexto, de tal forma que esta rota representa os sentidos mais relacionados no contexto.

Em experimento realizado com diversos conjuntos de testes, o método que mais se destacou foi a utilização de todas as métricas de similaridade. Posteriormente, os autores comparam este algoritmo com outros métodos da literatura e variações por meio de sistemas de votação. Os métodos utilizados são: (TSP-ACO, algoritmo de resolução do problema do CV por meio da abordagem de Colônia de Formigas), método com melhor resultado no experimento anterior; verificação do grafo de incidência no vértice (*synset*) presente no grafo de desambiguação; método de Agirre and Soroa (2009); e método de Lesk (1986).

Para estas comparações, utilizam três conjuntos de testes (Senseval-2, Senseval-3 e Senseval 2007). De forma isolada, sem mecanismo de votação, o método TSP-ACO apresenta-se superior ao método de Agirre and Soroa (2009) no primeiro conjunto de testes e ao método de Lesk no último conjunto, sendo inferior à *baseline* de sentido mais frequente em todos os casos.

Posteriormente, os autores realizam combinações entre os métodos para sistemas de votações: voto binário e contagem de pontos. A técnica de voto binário corresponde à verificação de qual *synset* foi mais selecionado pelos métodos. Por exemplo, sendo três *synsets* $\{s^1, s^2, s^3\}$ e três métodos de DLS $\{m^1, m^2, m^3\}$, admitindo que, para uma determinada PA, m^1 e m^2 selecionaram o *synset* s^1 e m^3 adotou s^2 , atribui-se o *synset* s^1 para a PA, pois mais métodos de DLS o selecionaram. Já na segunda abordagem de votação, sendo que cada método atribuiu uma pontuação para cada *synset*, seleciona-se aquele com maior soma de pontuações.

Dada as estratégias de votação supracitadas, foram adotadas as seguintes configurações: TSP-ACO + Agirre and Soroa (2009) + Lesk (voto binário); Grau de incidência + Agirre and Soroa (2009) + Lesk (voto binário); TSP-ACO + Agirre and Soroa (2009) + Lesk (quantidade de votos). Com estas combinações, ainda os três conjuntos de testes supracitados, a última variação mostrou-se superior, tendo valor máximo de 81,1% de precisão no último conjunto de testes.

Contudo, vale ressaltar que os autores não apresentam o método de votação TSP-ACO + Agirre and Soroa (2009) + Lesk por meio de contagem de votos, mas dispõem apenas deste método com votação binária. Assim, considerando que o método TSP-ACO apresenta resultados inferiores na maioria dos casos, o melhor método de votação apresentado pelos autores pôde apresentar melhores resultados ao tomar proveito da quantidade de votos de outros métodos, que também poderia ser adicionada no método apenas com votação binária.

3.2 Abordagens Baseadas em Córpus

Turney (2004) desenvolve um sistema de DLS que utiliza AM e informações de probabilidades de co-ocorrência das palavras. Seu sistema foi submetido à tarefa de *English lexical sample* (tarefa de *lexical sample* para o idioma Inglês) do Senseval-3⁵, que se constituía na desambiguação de 57 palavras do inglês e disponibilizava aos participantes um conjunto de 140 exemplos para treinamento de modelos de aprendizado e 70 exemplos para testes.

Para implementação do modelo de aprendizado, o autor utilizou o software Weka (Witten and Frank, 2005). As informações aplicadas para o aprendizado são classificadas pelo autor entre sintáticas e semânticas. As informações sintáticas são constituídas pela etiqueta morfossintática, determinada por meio de um *tagger* automático, e um atributo para indicar se a palavra está no plural ou singular.

As informações semânticas indicam a similaridade semântica entre a palavra alvo e as demais de seu contexto. A similaridade semântica é representada por um valor real, entre 0 e 1, que estipula quão próximas semanticamente são duas palavras por meio de conceitos utilizados em pesquisas na recuperação de informação (Turney, 2004).

O autor compara seu sistema com o melhor sistema do Senseval-3 e com a *baseline* atribuída pelo sentido mais frequente na Wn-Pr. Seu sistema não superou o melhor ranqueado no Senseval-3, porém se mostrou sempre melhor do que os valores da *baseline*. O autor ressalta que o sistema proposto é uma abordagem clássica de AM, portanto, não esperava grandes resultados. Contudo, diz que o uso da similaridade semântica pode contribuir para melhores resultados.

Domingo and Roxas (2006) abordam a DLS de forma estatística visando aplicações para Tradução Automática. Sua aplicação de DLS está diretamente relacionada à forma na qual se propõem realizar tradução, empregando o conceito tradução por seleção, que corresponde a selecionar primeiro os sentidos das palavras e, posteriormente, traduzir as palavras por meio do significado.

⁵<http://www.senseval.org/senseval3>

Os autores empregam um modelo estatístico pautado na probabilidade de escolha do melhor *synset* na hierarquia da Wn-Pr. Esta probabilidade é adquirida por meio da utilização de um *córpus* com exemplos de traduções, composto por 317.113 palavras derivadas do Novo Testamento que receberam anotações sintáticas por meio de um *parser* sintático automático.

Os autores apresentam testes pautados na tarefa de tradução, apontando que tal método de DLS é melhor empregado para desambiguação de verbos, não apresentando testes mais elaborados ou melhor especificados quanto aos resultados do módulo de desambiguação.

Mihalcea (2007) apresenta uma proposta para criação de *córpus* anotados com sentidos utilizando recursos disponíveis na Wikipedia⁶. Posteriormente, aplica esta anotação na tarefa de DLS usando modelos de AM.

A utilização da Wikipedia para criação de *córpus* torna-se viável devido a sua organização de conteúdo e sua estrutura de *hyperlinks*. A Wikipedia é constituída por artigos, que são hipertextos formados por descrições de alguma entidade ou evento e referenciados por identificadores únicos. Outra característica relevante é a redundância de informações na Wikipedia, onde um mesmo conceito é referenciado em diferentes páginas. Por exemplo, o conceito *circuit (electric)*, que é referenciado por *circuit*, *integrated circuit*, *electric circuit*, e outros. Diferentes referências para um mesmo conceito sugerem nomes alternativos e possivelmente auxiliam a DLS, pois apresentam contextos de uso dessas palavras. Outro recurso disponível na Wikipedia são as páginas de desambiguação, que são criadas com termos ambíguos que possuem referências para diversos artigos.

Com as características supracitadas da Wikipedia, os autores apresentam um método para geração de *córpus* resumido em três etapas:

1. extração de todos os parágrafos que possuem uma palavra ambígua usada como link. Para impedir que o método extraia entidades nomeadas, somente palavras iniciadas com letra minúscula são aceitas;
2. coleta de todos os rótulos possíveis para uma palavra por meio da remoção do componente mais a esquerda dos links. Exemplo, do link *[[São Paulo (cidade)| São Paulo]]* extrai-se *São Paulo (cidade)*; e
3. mapeamento dos rótulos de forma manual para seus correspondentes *synsets* na Wn-Pr, por meio de anotação de *córpus*.

A etapa de anotação foi realizada por dois anotadores, sendo que esses anotavam os sentidos de forma independente. Quando havia divergências, o sentido correto era atribuído por um consenso entre os dois anotadores ou com auxílio de um terceiro anotador.

⁶<http://www.wikipedia.org>

A medida kappa ([Carletta, 1996](#)), que é uma métrica empregada para aferir o nível de acordo entre diferentes juizes em uma mesma tarefa descontando-se a concordância por acaso, foi de $k = 87.1$, o que representa um alto nível de concordância.

Após anotação do corpus, um modelo de aprendizado de máquina (*Naïve Bayes*) foi treinado com os seguintes atributos (P representa a palavra a ser desambiguada): classificação gramatical de P ; janelas de palavras com tamanho 3; verbos e substantivos antes e depois de P ; e contexto global gerado por meio dos sentidos das cinco palavras mais frequentes que ocorrem pelo menos três vezes nos contextos de definição de um sentido da palavra alvo.

Os experimentos foram avaliados para 30 palavras ambíguas, que eram substantivos associados a entidades definidas pela Wikipedia. Para comparação do método proposto, duas *baselines* foram aplicadas: o sentido mais frequente na Wn-Pr e o algoritmo de [Lesk \(1986\)](#). O método apresentado mostra-se inferior apenas em 4 casos: para as palavras *grip*, *nature*, *post* e *spade*.

Os autores apontam que a deficiência do método de DLS é devida às características do corpus utilizado. A construção do corpus pauta-se na estrutura de *links* da Wikipedia, mais precisamente aqueles formados por mesmas palavras e com identificadores diferentes. Algumas palavras não apresentam diferenças nos identificadores e outras não os possuem, contribuindo para poucos exemplos no corpus.

[Specia \(2007\)](#) apresenta um método de DLS voltado para a TA entre o Português e o Inglês. Seu método se propõe a desambiguação de 10 verbos do inglês (*ask*, *come*, *get*, *give*, *go*, *live*, *look*, *make*, *take* e *tell*), considerados mais ambíguos pela autora. A principal característica de sua abordagem é o uso de um modelo de programação denominado Programação Lógica Indutiva (PLI), que se caracteriza por técnicas de AM em conjunto com regras especificadas em lógica proposicional.

Para desenvolvimento de seu método de desambiguação, foi criado um corpus paralelo que possui textos em Inglês lexicalmente alinhados com suas respectivas traduções em Português. O corpus criado utilizou nove fontes de variados gêneros e domínios, totalizando 7.606.150 palavras em inglês e 7.642.048 em português⁷.

Em sua abordagem, a quantidade de significados dos verbos é dada pela quantidade de possíveis traduções, de acordo com dicionários bilíngues. Assim, quanto maior a possibilidade de traduções, mais ambíguo um verbo é. A quantidade de possíveis traduções foi de: *ask* = 16, *come* = 226, *get* = 242, *give* = 128, *go* = 197, *live* = 15, *look* = 63, *make* = 239, *take* = 331 e *tell* = 28.

O método apresentado por [Specia \(2007\)](#) utiliza várias fontes para extrair conhecimento. Cada uma destas foi definida em Prolog, usando sua representação de cláusulas

⁷A autora não diz se a contagem de palavras omitiu *stopwords*.

lógicas. A seguir, é apresentada a listagem destas fontes de conhecimento utilizadas por sua abordagem:

- *bag of words* para representação de contextos;
- bigramas frequentes: pares de palavras adjacentes em uma sentença que ocorrem pelo menos 10 vezes no *corpus*;
- classificação gramatical: classes gramaticais das 5 palavras em torno do verbo a ser desambiguado;
- contexto lematizado: lema das 5 palavras em torno do verbo a ser desambiguado;
- colocações: variações morfológicas relevantes do verbo;
- relações sintáticas;
- relações sintáticas frequentes: pares de palavras sintaticamente relacionadas na sentença que ocorrem pelo menos 10 vezes no *corpus*;
- sobreposição de definições: o sentido (ou tradução) que apresenta a maior taxa de sobreposição entre as palavras na sua definição, semelhante à proposta de [Lesk \(1986\)](#);
- restrições de seleção: restrições definidas sobre os traços semânticos dos argumentos dos verbos;
- expressões verbais; e
- e regras criadas após observações e experimentos no *corpus* paralelo empregado.

A autora apresenta tanto avaliações intrínsecas quanto extrínsecas para seu método. Porém, são apresentadas a seguir apenas as avaliações intrínsecas, pois métodos que avaliam sistemas de TA não são o foco deste trabalho. Os testes realizados usaram dois conjuntos dos verbos apresentados anteriormente: o primeiro formado por 7 verbos (todos os 10 exceto *ask*, *live* e *tell*) e o segundo formado por todos os 10 verbos. A *baseline* empregada foi o sentido mais frequente e algumas outras abordagens de AM, como é apresentado a seguir.

O primeiro experimento destinou-se apenas ao conjunto de 7 verbos. As fontes de conhecimento utilizadas foram: *bag-of-words*, etiquetas gramaticais, colocações, relações sintáticas de sujeito e objeto, sobreposição de definições, restrições de seleção e expressões verbais. Como valores de *baseline*, além do sentido mais frequente, fez uso de dois algoritmos implementados na ferramenta Weka, o algoritmo J48 (abordagem de árvore de

decisão) e o algoritmo SMO (abordagem de SVM). O valor médio de acurácia obtido pelo método foi de 75% contra 56% do sentido mais frequente, 61% do algoritmo J48 e 67% do algoritmo SMO. Apenas a desambiguação do verbo *give* apresentou resultados inferiores à *baseline* do sentido mais frequente, 96% contra 98%.

O segundo experimento assemelha-se bastante com o primeiro, exceto que, foi utilizada a fonte de conhecimento *narrow context* e o conjunto de testes possuía exemplos negativos gerados artificialmente. O valor médio de acurácia obtido pelo método foi de 78% contra 56% do sentido mais frequente, 60% do algoritmo J48 e 65% do algoritmo SMO. Nessa configuração, o método apresentou uma melhora de 3% para o primeiro experimento e, não se mostrou inferior a nenhum valor de *baseline*.

O terceiro experimento fez uso das mesmas fontes de conhecimento anteriores. Seu objetivo foi avaliar variações do algoritmo SMO. A primeira configuração do algoritmo utilizou as seguintes fontes de conhecimento: *bag-of-words*, etiquetas gramaticais, colocações, relações sintáticas de sujeito e objeto e sobreposição de definições. A segunda configuração utilizou 5000 atributos gerados por meio da PLI, porém, convertidos para atributos binários. Já a terceira configuração, foi constituída por todos os atributos da primeira e da segunda variação após passar por uma etapa de seleção de atributos, também implementada no software Weka. Os valores médios de acurácia obtidos, em ordem de configuração, foram de 67%, 73% e 78% contra 56% da *baseline*. O único verbo que apresentou resultado inferior ao da *baseline* foi o verbo *give*, 95% a segunda variação contra 98% da *baseline*.

O quarto experimento utiliza todos os 10 verbos e inclui todas as fontes dos experimentos anteriores adicionando: bigramas que ocorrem com alta frequência no corpus, pares de palavras relacionadas sintaticamente e duas fontes específicas para a tradução, *bag-of-words* do português e *narrow context* do português. Os modelos avaliados neste experimento foram o J48, Naïve Bayes, SMO, seu método de PLI e uma variação deste com ordenação das regras de forma manual. Os resultados apresentaram que os modelos de PLI foram superiores em todos os casos, com valores médios de acurácia de 72% da variação do modelo em PLI, 74% do modelo original de PLI, 68% do SMO, 65% do *Naïve Bayes*, 59% do J48 e 50% do sentido mais frequente.

Os experimentos 5 e 6 foram executados sobre o corpus do Senseval-3. Os métodos avaliados foram o modelo original de PLI e o modelo com ordenação das regras, que também foram aplicados no experimento 4. Os valores de *baseline* utilizados foram o sentido mais frequente e o algoritmo SMO. Os valores médios de acurácia obtidos foram de 72% para o modelo de PLI modificado, 69,2% do modelo original de PLI, 68,6% do algoritmo SMO e 55% do sentido mais frequente.

O experimento 7 foi destinado a testar uma otimização do modelo de PLI empre-

gado no experimento 5. Essa otimização foi realizada pelo uso das seguintes fontes de conhecimento: *bag-of-words*, bigramas, etiquetas gramaticais, *narrow context*, colocações, relações sintáticas de sujeito e objeto, pares de palavras relacionadas sintaticamente, sobreposição de definições e restrições de seleção. O modelo proposto foi empregado para a desambiguação de 32 verbos do conjunto de testes do Senseval-3. O valor médio da acurácia alcançada foi de 72% contra 55% da *baseline* do sentido mais frequente.

Os experimentos 8 e 9 foram realizados pelo conjunto de testes da tarefa de *lexical sample* do Senseval-3. Esta tarefa constitui-se pela desambiguação de 100 palavras, sendo 65 verbos e 35 substantivos. No experimento 8 foram utilizadas as seguintes fontes de conhecimento: *bag-of-words*, bigramas, etiquetas gramaticais, *narrow context*, colocações, relações sintáticas de sujeito e objeto, pares de palavras relacionadas sintaticamente, sobreposição de definições, restrições de seleção e expressões verbais (utilizada apenas para desambiguação de verbos). O experimento 9 fez uso dos mesmos dados do experimento 8, porém, usando o algoritmo SMO. Os valores médios de acurácia obtidos pelos métodos para desambiguação de verbos foram, em ordem crescente dos experimentos, 81,7% e 82,8% contra 76,2% da *baseline* do sentido mais frequente. Para desambiguação de substantivos, os valores médios de acurácia foram de 88,2% para ambos os métodos contra 80,9% da *baseline*.

Che and Liu (2010) apresentam um modelo representado em Lógica de Markov que agrega à tarefa de DLS a rotulagem de papéis semânticos (ou SRL, do inglês *semantic role labeling*) e vice-versa. Para os autores, estas tarefas em PLN são de grande importância e possuem grande relação, portanto, devem ser tratadas em conjunto.

Nesse âmbito, os autores propõem um sistema em *pipeline* a fim de apresentar o modo como a DLS pode auxiliar o processo de Rotulagem de Papéis Semânticos (RPS), bem como a maneira como a anotação de papéis semânticos podem auxiliar a desambiguação. Para avaliar o trabalho, os autores também desenvolveram modelos que trabalham de forma isolada, tanto para DLS quanto para RPS e, posteriormente, comparam seus resultados com o sistema em *pipeline*.

O sistema é composto por um conjunto de predicados especificados em Lógica de Markov – descrita por expressões em lógica de primeira ordem, às quais são atribuídas pesos – para representar os sentidos e regras semânticas atribuídas a cada item lexical.

Os sentidos são representando pelo predicado $sense(w, s)$, indicando que uma palavra w possui o significado s . O predicado $role(p, a, r)$ representa as regras semânticas, onde o argumento na posição a é regido por uma regra r relacionada a um predicado na posição p . Também especificam o predicado $lemma(i, l)$, que indica que uma palavra na posição i possui o lema l .

Seus experimentos utilizam o *córpus* OntoNotes Release⁸. A utilização foi de 60% para treinamento do modelo, 20% para desenvolvimento e 10% para testes. Para realização de testes, os autores empregaram como *baseline* o sentido mais frequente na Wn-Pr, que obteve um valor 85,58% de acurácia. Após realização dos experimentos, os métodos testados obtiveram sempre acurácia 89,40%, que foi um pouco superior ao valor da *baseline*.

Silva and Amancio (2012) apresentam métodos de DLS por meio da utilização de classificadores de alta ordem e redes complexas. As redes complexas são empregadas para capturar padrões de contextos entre as palavras a serem desambiguadas e as palavras que as circundam em uma sentença ou texto. Posteriormente, aplicam algoritmos de Aprendizado de Máquina para treinar classificadores sobre os padrões representados pelas redes.

Para tanto, fazem uso de dois níveis de classificação. No primeiro nível, utilizam variações com os algoritmos de aprendizado de máquina C4.5, kNN e *Naïve Bayes*. No segundo nível de classificação, verificam os padrões presentes na rede complexa. A união desses classificadores é dada por um fator α , que indica pesos para ambos. Ou seja, dados dois classificadores, c_1 e c_2 , a classificação é dada pelos pesos $(1 - \alpha)c_1 + \alpha c_2$.

Os autores avaliam seus métodos na tarefa de *lexical-sample*, com as palavras do inglês *save*, *note*, *march*, *present* e *jam*. A melhor combinação dos métodos (primeiro e segundo nível) foi dada pela utilização do kNN, que se apresentou superior para quatro palavras. Ressalta-se que o valor de α adotado, para este algoritmo, foi sempre o menor, sendo zero em dois casos (*present* e *jam*). Ou seja, os resultados alcançados nessa variação indicam uma menor utilização do classificador de segundo nível.

A única palavra para a qual a combinação de classificadores supracitada mostrou-se inferior foi *present*. Nesse caso, o melhor classificador foi o que utilizou somente o algoritmo *Naïve Bayes*. É importante também ressaltar que o valor de α utilizado foi zero, que indica, assim como no caso anterior, baixa melhoria na utilização do classificador de segundo nível.

3.3 Síntese dos Trabalhos Discutidos

Abordagens para métodos de DLS possuem grandes distinções, como: conjunto de palavras a ser desambiguado, idioma, recursos utilizados, repositório de sentidos, tarefas designadas, etc. A utilização de documento Web como *córpus* vem sendo investigada por alguns trabalhos, tais como: (Mihalcea and Moldovan, 1999), que fazem uso de ferramentas de busca na web; (Mihalcea, 2007), que utilizam documento da Wikipedia, para criação de exemplos de palavras ambíguas; e Nguyen and Ock (2011), que empregam

⁸<http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?>

informações da Wikipedia para adicionar mais informações às glosas dos *synsets*.

Recentemente, abordagens baseadas em grafos passaram a ser empregada na DLS. Essas abordagens visam aproveitar as características hierárquicas de alguns RS, como *wordnets* e ontologias, o que não seria possível por meio da utilização de dicionários comuns, que não possuem relações entre os sentidos. Contudo, métodos de DLS dessa abordagem, que, normalmente, aplicam algoritmos de ranqueamento em grafos, se limitam a um sentido por palavra, ou seja, todas as ocorrências de uma mesma palavra em determinado contexto são desambiguadas com um mesmo significado.

O cenário supracitado contrasta com os sistemas do Estado da Arte na área, composto pela abordagem de Aprendizado de Máquina. Entretanto, ressalta-se que as comparações são realizadas em domínios bem especificados. Acredita-se que tais abordagens atinjam melhores resultados por tratarem domínios específicos. Por outro lado, abordagens baseadas em conhecimento tornam-se mais abrangentes, pois podem, facilmente, serem empregadas para desambiguar um número maior de palavras.

É apresentada no Quadro 3.1 uma sumarização dos trabalhos relacionados citados neste capítulo. A ordenação das linhas é dada pelo ano do trabalho. As colunas representam, respectivamente: nome dos autores e ano de publicação; tipo de abordagem; recursos utilizados; e idioma que o método é capaz de tratar.

A fim de atingir os objetivos propostos nesta dissertação, foram selecionados, principalmente, trabalhos de DLS da abordagem baseado em conhecimento, enfatizando os mais citados na literatura. Trabalhos baseados em algoritmo de AM também foram citados, isso porque se buscou conhecer trabalhos com maior desempenho em relação aos da abordagem anterior. É importante ressaltar que existem diversos outros trabalhos relevantes na área, que são apresentados a seguir.

Black (1988) apresenta um algoritmo de DLS baseado em árvores de decisão. Para tanto, por meio de testes empíricos e investigação de *corpus*, codifica regras de decisão de forma manual. Hearst (1991) apresenta, possivelmente, o primeiro método de desambiguação baseado em técnicas semi-supervisionadas.

Wilks and Stevenson (1997) apresentam o conceito de DLS em cascata, quando se utilizam de vários métodos de desambiguação, considerados simples, em sequência. Os autores partem do princípio de redução de dimensionalidade por meio de camadas de desambiguação, ou seja, para uma determinada palavra p com n possíveis sentidos, o objetivo da primeira camada (primeiro método de DLS) não é selecionar o sentido mais adequado, mas descartar alguns que considera incorretos e, assim, sucessivamente, a cada camada, as possibilidades de desambiguação são diminuídas. Portanto, a desambiguação propriamente dita é determinada somente na última camada.

Por fim, tem-se o trabalho de Veronis (1998), que, possivelmente, foi o primeiro tra-

balho de DLS a fazer uso de Redes Neurais.

Quadro 3.1: Trabalhos de DLS

Trabalho	Abordagem	Recursos	Idioma
Lesk (1986)	Conhecimento	Dicionários	Inglês Independente
Mihalcea and Moldovan (1999)	Conhecimento	Buscadores Web	Inglês
Banerjee and Pedersen (2002)	Conhecimento	Adaptação do método de Lesk (1986)	Inglês
Turney (2004)	AM	Córpus anotado	Inglês
Vasilescu et al. (2004)	Conhecimento	Variações no Método de Lesk	Inglês
Domingo and Roxas (2006)	Estatística	Córpus paralelo	Inglês– Alemão
Mihalcea (2007)	AM	Wikipedia Córpus anotado	Inglês
Sinha and Mihalcea (2007)	Conhecimento	Grafo Similaridade	Inglês
Specia (2007)	AM, Regras	Córpus paralelo	Português– Inglês
Agirre and Soroa (2009)	Conhecimento	Grafos PageRank	Inglês
Che and Liu (2010)	Lógica de Markov	Córpus Regras manuais	Inglês
Machado et al. (2011)	Ontologias Desambiguação geográfica	Baseado em conhecimento	Independente
Nguyen and Ock (2011)	Conhecimento	Grafo Otimização	Inglês
Silva and Amancio (2012)	AM	Redes complexas Alta ordem	Inglês

Na Figura 3.1, são apresentados os trabalhos supracitados em uma representação temporal, sendo possível a visualização dos pontos relevantes de cada trabalho, bem como a distribuição destes. Para formação desta representação, foram utilizados os trabalhos referenciados neste texto e recursos importantes para a área, como a WordNet de Princeton.

Figura 3.1: Trabalhos de DLS

Anotação de Corpus

Como dito anteriormente, na subseção de objetivos, o corpus adotado neste trabalho foi o CSTNews, por esse ser constituído por textos jornalísticos (que apresentam uma linguagem mais comum e que, possivelmente, métodos de DLS avaliados nesse cenário possam ser mais facilmente empregados em outros contextos, principalmente em páginas da Web). Tais justificativas e mais detalhes do corpus encontram-se na Seção 4.1.

Neste capítulo, será apresentado o processo de anotação do corpus CSTNews, bem como os resultados e análise dos mesmos. Para tanto, tem-se a seguinte estrutura: Na Seção 4.1, será tratado o processo de anotação do corpus, bem como a metodologia de anotação empregada neste processo (Subseção 4.1.1) e a ferramenta de anotação desenvolvida (Subseção 4.1.2).

A análise de concordância entre os anotadores e seu respectivo processo será disposta na Seção 4.2. Por fim, na Seção 4.3, disserta-se sobre os resultados e análises dos dados anotados.

4.1 Anotação do Corpus

A anotação constituiu em desambiguar somente palavras etiquetadas como substantivos comuns, limitando-se a apenas 10% dos substantivos mais frequentes da coleção de textos. A opção por somente palavras desta classe morfossintática respalda-se em trabalhos da literatura, como Plaza and Diaz (2011), que, por meio de experimentos para avaliação de métodos de Sumarização Automática, relataram que a desambiguação de substantivos contribuiu mais significativamente para melhorias na Sumarização e, pos-

sivelmente, outras aplicações do PLN. Ao verificarem o impacto da desambiguação de outras classes morfossintáticas, como verbos, adjetivos e advérbios, relatam resultados pouco satisfatórios.

Para complementar a justificativa supracitada, foi realizado um experimento para avaliar a distribuição das classes gramaticais no corpus CSTNews. Para tanto, aplicou-se o *tagger* MXPOST em todos os textos e, após esta etapa, computou-se a frequência de ocorrência das etiquetas. Esta distribuição é ilustrada na Figura 4.1, onde se pode observar que a etiqueta de substantivos comuns corresponde a mais de 50% do corpus.

Figura 4.1: Percentual de ocorrência das etiquetas morfossintáticas no corpus CSTNews (Somente classes gramaticais abertas)

Inicialmente, o objetivo da anotação era desambiguar todos os substantivos comuns presentes na coleção de textos. Porém, após a etapa de treinamento dos anotadores (que será apresentada na Subseção 4.1.1), optou-se por diminuir o escopo da tarefa, visto que foram necessárias 3 reuniões para desambiguar apenas um texto dentre três de uma coleção. Assim, o objetivo da tarefa foi a desambiguação dos substantivos mais frequentes em cada coleção, limitando-se a 10% do total de substantivos presentes em todos os textos da coleção. Esse limiar foi adotado, pois foi observar que palavras abaixo desse limiar ocorrem muito pouco nos textos e, consequentemente, foram consideradas pouco representativas para a tarefa.

A amostra anotada foi suficiente para verificar os objetivos deste trabalho de mestrado. Por exemplo, por meio desta pôde-se verificar a distribuição de sentido por palavra no cenário multidocumento, bem como avaliar os métodos de DLS, mono e multidocumento, desenvolvidos por meio da abordagem multilíngue proposta.

Esta tarefa foi desenvolvida com um grupo de 10 anotadores, composto por alunos de pós-graduação (mestrado e doutorado) e professores do Instituto de Ciências Matemáticas e de Computação – USP e Universidade Federal de São Carlos. Foram necessárias seis semanas para o término da anotação, sendo uma destas destinada ao treinamento do

grupo.

4.1.1 Metodologia de Anotação

Para auxiliar os anotadores, foi desenvolvida uma ferramenta de anotação que automatiza partes da metodologia de anotação, intitulada NASP (NILC – Anotador de Sentidos para o Português). O processo de anotação ocorreu em reuniões diárias com uma hora de duração, sendo necessárias seis semanas para o término. A primeira semana foi destinada ao treinamento do grupo de anotadores, que foi composto pelas seguintes etapas: 1) apresentação da ferramenta de anotação NASP (que será apresentada a seguir) e metodologia da tarefa aos anotadores; 2) criação de grupos, entre os anotadores, com dois ou três membros; e 3) para cada grupo, anotação de todos os substantivos comuns presentes nas coleções, com dois ou três textos cada (tais coleções de textos encontram-se no Apêndice A). As demais semanas (cinco semanas no total) foram empregadas na anotação propriamente dita, sendo três dias (não sequenciais) destinados às anotações para avaliação de concordância.

Cada coleção de textos do CSTNews foi anotada uma única vez por um determinado grupo de anotadores, exceto as coleções que foram empregadas para aferir valores de concordância e, portanto, foram anotadas por todos os grupos.

O agrupamento dos anotadores era realizado de tal forma a unir membros com diferentes níveis (alunos e professores), diferentes áreas (linguistas e cientistas da computação) e que com menos frequência pertenceram a um mesmo grupo. Tais critérios foram empregados visando nivelar o conhecimento dos grupos e, de certa forma, não “viciar” a tarefa (que pode ocorrer quando um mesmo grupo é formado várias vezes em dias seguidos), buscando assim, manter a qualidade do processo.

A metodologia de anotação pode ser subdividida em duas partes: metodologia geral e individual. Por metodologia geral, entendem-se as etapas que os grupos de anotação deveriam seguir para anotar toda a coleção de textos. Já a metodologia individual foi destinada à descrição do processo de desambiguação de uma palavra individualmente.

A metodologia geral foi empregada a fim de tirar proveito do cenário multidocumento presente no corpus, partindo do princípio heurístico que uma palavra tende a assumir um mesmo significado em um mesmo contexto (sentença, texto, etc.) (Mihalcea, 2006). Para tanto, empregou-se o mecanismo de palavras “pré-anotadas”, que corresponde as palavras cujas ocorrências anteriores (referenciadas pela forma lematizada) já tenham sido previamente anotadas.

Por exemplo, havendo n ocorrências da palavra “banco”, e sendo sua primeira ocorrência anotada com o *synset* s e tradução t , todas suas demais ocorrências, ainda não anotadas, terão os mesmos *synsets* e tradução. Assim, auxiliado pela NASP, os grupos de

anotadores deveriam confirmar, alterar ou excluir estas anotações previamente estipuladas.

A vantagem da utilização deste mecanismo foi, principalmente, a diminuição do tempo de anotação. Isto porque, os anotadores poderiam, primeiramente, verificar se uma palavra pré-anotada necessitaria de alteração de sentido, antes mesmo de selecionar possíveis traduções e, posteriormente, avaliar uma lista de *synsets*. Do ponto de vista técnico, a utilização de palavras pré-anotadas contribuiu para a redução de chamadas à API da ferramenta de tradução, o WordReference®¹, que foi empregada no software de anotação.

Do ponto de vista teórico, partindo da hipótese que uma palavra tende a manter um mesmo significado em uma coleção de documentos, que foi posteriormente validada por meio de verificação de *corpus* (Seção 4.3), esse mecanismo mostrou-se útil por corresponder às características multidocumento da anotação. Ou seja, a utilização de palavras pré-anotadas foi eficiente, pois, no CSTNews, as palavras, em geral, apresentam apenas um sentido na coleção.

Nesta metodologia, empregou-se a seguinte sequência de etapas: 1) dada uma coleção de textos, escolher um texto para ser anotado (era indicado iniciar pelo menor, fazendo com que os textos maiores tenham uma maior quantidade de palavras pré-anotadas); 2) anotar todas as palavras assinaladas pela ferramenta NASP e, somente após anotar todas as palavras de um texto, anotar outro texto da coleção; 3) avaliar as palavras pré-anotadas para confirmar, alterar ou excluir tais anotações; e 4) após anotar todos os textos, revisar e salvá-los no formato especificado (também auxiliado pela ferramenta NASP).

A metodologia individual, destinada à anotação de cada palavra, pode ser observada na Figura 4.2. Primeiramente, a palavra alvo deve ser traduzida para o inglês. Para tanto, a ferramenta NASP apresenta uma listagem das possíveis traduções por meio do WordReference®. Após se selecionar a melhor tradução (que pode ser manualmente inserida, seja por ausência ou erro de tradução automática), a ferramenta lista os possíveis *synsets* para a respectiva seleção. Assim, cabe ao anotador escolher um sentido dentre os listados ou escolher uma nova tradução.

Para palavras pré-anotadas, esta metodologia ainda é válida. Contudo, nestes casos, primeiramente, o sistema NASP apresenta a possível anotação e cabe ao anotador concordar ou não. Caso concorde, basta confirmar o *synset* selecionado, com isto, a respectiva tradução é salva automaticamente pelo sistema. Caso não concorde, utiliza-se normalmente do processo individual de anotação.

Na próxima subseção, será apresentada a ferramenta NASP, bem como mais alguns detalhes de sua utilização durante o processo de anotação.

Figura 4.2: Metodologia de Anotação

4.1.2 Ferramenta de Anotação: NASP

A ferramenta NASP, como principais características, apresenta: pré-processamento da coleção de textos; tradução das palavras por meio do tradutor automático WordReference®; recuperação dos possíveis *synsets* de uma palavra após selecionar uma tradução; gerenciamento das palavras anotadas, não anotadas e pré-anotadas; e gerenciamento do formato de anotação.

A etapa de pré-processamento possui o objetivo de selecionar as palavras a serem anotadas e prepará-las para a mecânica de palavras pré-anotadas. Assim, têm-se as seguintes etapas: tokenização dos textos; aplicação do etiquetador morfosintático MXPOST, que somente é viável após a primeira etapa; lematização das palavras por meio do lematizador disponível pelo grupo de pesquisa NILC¹, para referenciá-las e permitir suas posteriores traduções; computação, por meio do lema e etiquetas, das frequências das palavras; e, por fim, seleção das palavras que se encontram entre as 10% mais frequentes.

Na Figura 4.3, é apresentada a tela principal da NASP. Esta ferramenta foi desenvolvida para anotação de sentidos no cenário multidocumento. Portanto, permite abrir um ou mais textos para serem manipulados. No segundo caso, a ferramenta os interpreta como uma mesma coleção.

¹Disponível em <http://www2.icmc.usp.br/~taspardo/LematizadorV2a.rar>

Figura 4.3: NASP (Nilc – Anotador de Sentidos para o Português)

A NASP é composta, basicamente, por quatro componentes: 1) barra de menus, localizada na parte superior; 2) painel de visualização de textos, que se apresenta na parte superior do canto esquerdo; 3) painel de seleção de tradução, disposto na parte inferior, logo abaixo do componente anterior; e 4) painel de seleção de *synsets*, que ocupa o lado direito da ferramenta. Na barra de menus, o usuário possui acesso às funcionalidades de entrada e saída, ou seja, carregar e salvar arquivo. No primeiro caso, a ferramenta permite a leitura de arquivos textos (txt) ou XML, que representam anotações efetuadas por meio da NASP.

No segundo componente, a ferramenta dispõe os textos carregados em abas diferentes, de tal forma que seja possível visualizar somente um destes por vez. Caso seja carregado apenas um texto, somente uma aba é visualizada. O conteúdo dos textos é organizado por meio de componentes visuais, que permitem a interação do usuário por meio de cliques e também gerenciam o estado de anotação. Ou seja, a ferramenta permite que o usuário selecione as palavras a serem anotadas e apresenta se estas se encontram não anotadas, anotadas ou pré-anotadas.

É importante ressaltar que os componentes que representam as palavras dos textos são organizados em *selecionáveis* e *não selecionáveis*. Os itens *não selecionáveis* são

aqueles que representam itens lexicais que possuem etiquetas de números ou sinais de pontuação, bem como palavras consideradas *stop words*. Os demais itens são considerados *selecionáveis* e podem assumir quatro estados diferentes:

- não anotado: itens que correspondem a palavras que não foram anotadas nem selecionadas para anotação. Contudo, caso necessário, podem ser selecionadas. Por exemplo, a ferramenta pode apresentar erros de etiquetagem ou de lematização e, assim, uma palavra pode não ser selecionada automaticamente no processo;
- selecionado e não anotado: itens que referenciam palavras que foram selecionadas para anotação, mas ainda não tiveram tradução e *synset* especificados. Para estes itens, são utilizadas bordas na cor vermelha para indicá-los;
- pré-anotado: itens que indicam palavras pré-anotadas. Estes itens são demarcados com bordas e um *background* na cor amarela; e
- anotado: elementos que indicam palavras que já foram anotadas. Estes componentes são caracterizados por bordas na cor verde.

No painel de seleção de tradução, listam-se as possíveis traduções das palavras selecionadas por meio de cliques no painel supracitado. Para não restringir este processo somente às possibilidades dispostas pelo dicionário WordReference®[®], este componente também permite intervenção manual, ou seja, permite a inserção de traduções de forma manual. Este recurso é muito útil quando não há opções de traduções ou as possibilidades listadas estão incorretas. É importante informar que sempre que o usuário seleciona uma palavra pré-anotada ou anotada, este componente apresenta a respectiva tradução adotada anteriormente.

Por fim, o painel de seleção de *synsets* é empregado para listar todos os possíveis *synsets* acionados por uma tradução. Para auxiliar o processo de escolha do sentido, também é permitido configurar as informações a serem visualizadas, adicionando a listagem de hipônimos e hiperônimos. Por padrão, somente o conjunto de sinônimos e a glosa são indicados. Assim como o painel de tradução, sempre que o usuário selecionar uma palavra pré-anotada ou anotada, é apresentado o respectivo *synset* adotado para esta.

A seguir, será apresentada uma breve sequência de utilização da ferramenta, na qual exemplifica-se o processo de anotação por meio do cluster C1 do corpus CSTNews. Primeiramente, os três textos da coleção passam pela etapa de pré-processamento, e, assim, somente as palavras que estão entre as 10% mais frequentes são demarcadas como *selecionadas e não anotadas*, como pode ser observado na Figura 4.4.

Neste exemplo, ainda na Figura 4.4, é ilustrado o processo de desambiguação da palavra “avião”. No painel de seleção de tradução, pode-se observar a listagem de possíveis

Figura 4.4: Exemplo da NASP: anotando uma palavra.

traduções. Após se selecionar a tradução “*plane*”, são apresentados todos os respectivos *synsets*. Especificamente neste caso, adotou-se o primeiro *synset* (*airplane*, *aeroplane*, *plane*).

Na Figura 4.5, é apresentada a seleção de uma palavra pré-anotada. Nesse caso, pode-se observar que: 1) a palavra “avião” da primeira sentença, cuja anotação foi ilustrada na imagem anterior, encontra-se com uma borda verde², que indica que esta foi anotada; 2) a segunda palavra “avião”, que está sob o ponteiro do mouse na tela, é destacada por meio de um fundo amarelo; 3) a palavra pré-anotada “avião” encontra-se com a mesma tradução e *synset* da anotação apresentada na Figura 4.4; e 4) outras palavras, que ainda não foram anotadas, encontram-se ainda marcadas com uma borda vermelha.

Esta ferramenta permite a gravação do estado da anotação por meio de dois formatos, texto (TXT) e linguagem de marcação (XML). Por padrão, apenas arquivos XML são gerados e são automaticamente gravados pela ferramenta. Para gravação do tipo texto, é gerado um arquivo para cada documento carregado na ferramenta. Já no formato XML, um arquivo encapsula todos os documentos anotados.

No formato TXT, os itens lexicais são tokenizados, separados por um caractere de espaço, e cada sentença do documento anotado é disposta separadamente em uma linha. Todos os itens lexicais são marcados com informações extras, demarcadas entre os

²Em caso de impressão não colorida, pode-se observar que itens demarcados com bordas verdes possuem bordas mais claras que os itens de borda vermelha. Neste exemplo, a única palavra com borda verde é “avião” na primeira linha.

Figura 4.5: Exemplo da NASP: selecionando uma palavra pré-anotada.

símbolos < e >. Para itens lexicais não anotados, entre os símbolos de marcação, é indicada somente a etiqueta morfofossintática. Para palavras anotadas, esta marcação segue o seguinte padrão:

<tag, manual?, traduções, synsets>

sendo: *tag* a etiqueta morfofossintática atribuída ao item lexical por meio do MXPOST; *manual?* um valor binário para representar o tipo de tradução, sendo 0 para traduções selecionadas dentre as opções listadas pelo WordReference® e 1 para tradução manualmente inserida; traduções é um conjunto com todas as possíveis traduções listadas para a palavra, sendo a tradução anotada listada como primeiro elemento do conjunto; e synsets o conjunto de todos os possíveis *synsets*, sendo o primeiro do grupo, o *synset* adotado.

Por exemplo, para a palavra “avião”, que foi empregada na exemplificação de ferramenta NASP, tem-se a seguinte marcação: `avião<N,0,{plane,airplane},{Noun@2691156[airplane, aeroplane, plane] - an aircraft that has a fixed wing and is powered by propellers or jets, Noun@13861050[plane, sheet] - (mathematics) an unbounded two-dimensional shape, Noun@13941806[plane] - a level of existence or development, Noun@3955296[plane, planer, planing machine] - a power tool for smoothing or shaping wood, Noun@3954731[plane, carpenter's plane, woodworking plane] - a carpenter's hand tool with an adjustable blade for smoothing or shaping wood}>`.

É importante ressaltar que se assume uma representação textual dos *synsets*, que corresponde a: classe gramatical @ número de identificação do *synset* [conjunto de palavras

```

-<save>
  -<Anotadores>
    <Anotador id="NÚMERO">NOME</Anotador>
    ...
  </Anotadores>
  -<Arquivos>
    -<Texto name="NOME DO ARQUIVO">
      -<p count="Nº DO PARÁGRAFO NO TEXTO">
        -<Token>
          <Valor>ITEM LEXICAL</Valor>
          <Etiqueta>ETIQUETA</Etiqueta>
          <Lema>LEMMA</Lema>
          <Tipo>ESTADO DE ANOTAÇÃO (0,1,2,3,4)</Tipo>
          -<Traducoes traducao_manual="TIPO DE TRADUÇÃO (0/1)">
            <Traducao selecionado="true">PRIMEIRA TRADUCAO</Traducao>
            <Traducao selecionado="false">SEGUNDA TRADUCAO</Traducao>
            ...
          </Traducoes>
          -<Synsets>
            <Synset selecionado="true">PRIMEIRO SYNSET</Synset>
            <Synset selecionado="false">SEGUNDO SYNSET</Synset>
            ...
          </Synsets>
        </Token>
        ...
      </p>
    </Texto>
    ...
  </Arquivos>
</save>

```

Figura 4.6: Hierarquia de Etiquetas XML da Ferramenta NASP

sinônimas | – glosa. Assim, para a palavra “avião” supracitada, tem-se que a interpretação de sua anotação é: um substantivo comum que foi traduzido como *plane*, cuja tradução foi listada automaticamente pelo dicionário bilíngue, e anotada com o *synset* da classe gramatical de substantivos (NOUN) e de número 2691156 da Wn-Pr.

O formato XML é constituído por um conjunto de marcas (itens) hierarquicamente organizado. Esse conjunto e sua ordenação são ilustrados na Figura 4.6. Pode-se dividir esta hierarquia em quatro principais níveis: Anotadores, Arquivos, Parágrafos (p) e Tokens. O primeiro nível, anotadores, é utilizado apenas para listar o nome dos integrantes do grupo de anotação.

A marcação *Arquivos*, no segundo nível, refere-se aos documentos anotados, que são, respectivamente, representados por etiquetas *Textos*, que possuem um atributo para identificar o título do documento. Este item é discretizado em parágrafos (marcação *p*), que são numerados pela respectiva ordem nos documentos e são compostos por *Tokens*, que encapsulam os itens lexicais anotados.

O item *Tokens* é composto por seis marcações, que representam informações diferentes do item lexical anotado, sendo estas: 1) *Valor*, o item lexical em sua representação no documento; 2) *Etiqueta*, a etiqueta morfofossintática atribuída; 3) *Lema*, a sua representação lematizada; 4) *Tipo*, um valor numérico que representa o estado da anotação (0 = item lexical não anotado; 1 = substantivo comum ainda não anotado; 2 = palavra anotada;

3 = item lexical que não pode ser anotado; e 4 = palavra previamente anotada); 5) *Traduções*, a listagem de todas as possíveis traduções para a palavra anotada; e 6) *Synsets*, a enumeração dos possíveis *synsets* para uma palavra.

Ressalta-se que para os itens *Traduções* e *Synsets*, a fim de padronização, tem-se que o primeiro item do conjunto é o elemento selecionado para a desambiguação e, para melhor representação, os elementos que representam traduções e *synsets* adotados na desambiguação também são indicados por meio do atributo *selecionado=true*.

4.2 Avaliação de Concordância

Para avaliar o nível de concordância, foram utilizadas três coleções (C15, C29, e C50) do corpus CSTNews. Os valores foram aferidos por meio de 4 métricas: (M1) medida Kappa (Carletta, 1996) ; (M2) concordância total; (M3) concordância parcial; e (M4) concordância nula.

A kappa é uma métrica bastante aplicada para aferir níveis de concordância. Seu funcionamento baseia-se em indicar o nível de acordo entre diferentes juízes em uma mesma tarefa, porém, descontando a concordância por acaso, por sorte. O cálculo da kappa é representado pela seguinte fórmula:

$$K = \frac{P(A) - P(E)}{1 - P(E)} \quad (4.1)$$

sendo, $P(A)$ o número de vezes que os juízes concordaram, e $P(E)$ a quantidade de vezes que era esperado a concordância, ou seja, acerto por sorte ou acaso. Assim, a interpretação da fórmula é dada pela quantidade de vezes que houve acerto subtraindo-se o número de acertos por acaso. E, para assumir valores entre 0 e 1, divide-se este resultado pelo percentual de instâncias do conjunto de teste, exceto aquelas consideradas como acaso.

Para as demais métricas de avaliação, foi analisada, para cada instância, a quantidade de acordo entre e os grupos de anotadores. Assim, foram contabilizados três diferentes níveis de acordo: (1) todos os grupos concordavam na anotação de uma instância; (2) a maioria dos grupos concordavam (neste caso, como eram formados apenas três grupos, este nível de acordo foi computado quando somente dois grupos concordavam na anotação); e (3) todos os grupos apresentavam anotações diferentes para uma mesma instância. Assim, tem-se que as métricas foram aferidas da seguinte forma:

- concordância total:

$$\sum \frac{n^\circ \text{ de vezes que todos os grupos concordavam}}{\text{total de instâncias}} \quad (4.2)$$

- concordância parcial:

$$\sum \frac{n^{\circ} \text{ de vezes com acordo entre a maioria}}{\text{total de instâncias}} \quad (4.3)$$

- concordância nula:

$$\sum \frac{n^{\circ} \text{ de vezes com discordância total}}{\text{total de instâncias}} \quad (4.4)$$

Devido à idiossincrasia desta tarefa de anotação, que exige etapas de tradução devido à utilização da Wn-Pr para desambiguação de textos em PT-BR, a fim de avaliar o impacto desta tarefa multilíngue, aferiu-se concordância sobre os seguintes critérios: (1) concordância na tradução das palavras; (2) concordância na escolha do synset; e (3) concordância na escolha da tradução e respectivo synset.

Tabela 4.1: Valores de concordância

	Kappa	Total (%)	Parcial (%)	Nula(%)
Tradução	0,853	82,87	11,08	6,05
<i>Synset</i>	0,729	62,22	22,42	14,36
Tradução– <i>Synset</i>	0,697	61,21	24,43	14,36

Os valores de concordância obtidos são apresentados na Tabela 4.1. Pode-se observar que o valor Kappa obtido para concordância entre *synsets* anotados, principal característica desta tarefa de DLS, é superior à 0.7, atingindo um bom valor de concordância. Como era esperado, o valor da concordância para tradução é um pouco superior (0.853), visto ser considerada uma tarefa menos subjetiva que a própria DLS. O valor de concordância obtido para tradução e respectivo *synset* foi menor (0,697), o que, de certa forma, era esperado, pois diferentes traduções podem referenciar a um mesmo *synset* e sentidos diferentes podem ser referenciados por traduções distintas.

É importante também salientar que a concordância total mostrou-se sempre superior às demais (parcial e nula), assim como o valor de concordância nula sempre foi inferior às demais (total e parcial). Ou seja, mesmo com a utilização de recursos da língua inglesa para a tarefa de desambiguação em PT-BR, tem-se, entre os anotadores humanos na tarefa manual, uma boa margem de concordância, o que indica afirmativamente a hipótese de que é possível desenvolver métodos de DLS para o PT-BR aplicando RS destinados a outra língua.

Dentre as palavras anotadas para avaliar a concordância, algumas instâncias obtiveram discordância total, ou seja, palavras que os anotadores sempre divergiam quanto ao *synset* mais adequado. Essas palavras são dispostas no Quadro 4.1.

Quadro 4.1: Palavras com discordância total entre os anotadores

acordo	conclusão	corte	dados
desoneração	escândalo	estado	gasto
imposto	próximo	referências	renda
resultado	setor	suposta	tucanos (contexto político)
valor			

4.3 Resultado da Anotação

Foram anotadas 4366 instâncias de palavras, sendo 465 palavras, 519 traduções e 575 *synsets* diferentes. A quantidade de exemplos anotados de uma mesma palavra variou de 1 a 105, sendo a palavra “ano” a mais frequente e as palavras “terra”, “defeito”, “notícia”, “meta”, “reportagem”, “ministério”, “reverso”, “mudança”, “falto” e “gesto” as menos frequentes, com apenas uma ocorrência.

Avaliando todas as instâncias anotadas, sem restringir a uma mesma coleção de textos, teve-se que uma mesma palavra assumiu entre 1 e 5 *synsets*. Esta distribuição de sentidos por palavra pode ser observada na Figura 4.7, onde cada coluna representa o número de palavras que apresentaram a quantidade de *synsets* expressa na abscissa. Por exemplo, pode-se observar que 386 palavras apresentaram apenas um sentido anotado no corpus. De forma geral, nota-se que a quantidade de *synsets* é inversamente proporcional à frequência das palavras, ou seja, há mais palavras que possuem apenas um *synset* do que palavras que possuem cinco sentidos diferentes.

Figura 4.7: Distribuição de *synsets* por palavra no corpus

Avaliando-se as instâncias anotadas restringidas em suas respectivas coleções, ou seja, verificando o comportamento das palavras e sentidos no cenário multidocumento, tem-se que a variação de *synsets* diferentes para uma mesma palavra ficou entre 1 e 3. Na Figura 4.8, é apresentada esta distribuição, na qual se pode observar que a maioria das palavras (677, que corresponde a 93,12%) foram anotadas com apenas um *synset*. Tal distribuição

indica, pelo menos neste contexto, que utilizar a heurística de um sentido por contexto é promissora.

Figura 4.8: Distribuição de *synset* por palavra em coleção

A dificuldade desta tarefa de anotação pode ser avaliada pela quantidade de possíveis *synsets* para cada palavra a ser desambiguada, o que representa o nível de ambiguidade destas. Na Figura 4.9, é apresentada a distribuição de palavras (eixo das ordenadas) por número de possíveis *synsets* (eixo das abscissas). Por exemplo, pode-se observar que 42 palavras possuem 5 possibilidades de desambiguação (possíveis *synsets*).

É importante ressaltar que a quantidade de possíveis *synsets* para uma palavra foi computada automaticamente por meio do método de busca de sentidos empregado neste trabalho, onde os *synsets* de uma palavra no PT-BR são retornados por meio de suas respectivas opções de traduções para o inglês. Assim, 66 palavras não apresentaram *synsets*, sendo 53 destas por não terem traduções³ e 13 porque suas traduções (no máximo duas) não referenciaram nenhum *synset*.

Figura 4.9: Distribuição de possíveis *synsets* ordenados

³De forma automática pelo dicionário bilíngue WordReference®.

Analisando esta distribuição, pode-se observar que o nível de ambiguidade das palavras é decrescente, ou seja, há mais palavras com menos *synsets* do que palavras com mais possibilidades de sentidos. Contudo, em um total de 466 palavras diferentes, 361 apresentaram dois ou mais possíveis sentidos, e dentre estas, 196 encontraram-se com possibilidades de desambiguação acima de média encontrada no *corpus*, que é aproximadamente 6 *synsets* distintos. Neste cenário, tem-se que cerca de 77% das palavras são ambíguas (com mais de dois sentidos) e aproximadamente 42% são altamente ambíguas, pois possuem *synsets* a cima da média.

Dado a idiossincrasia da abordagem multilíngue de DLS proposta neste trabalho, que necessita de etapa de tradução automática, analisou-se a distribuição da quantidade de sentidos em relação às possibilidades de tradução das palavras. O gráfico desta análise é apresentado na Figura 4.10, onde pode-se observar que, embora tenha uma pequena oscilação da curva, há uma tendência que a quantidade de *synsets* é diretamente proporcional às possibilidades de tradução, ou seja, quanto mais possíveis traduções há para uma palavra, mais ambígua esta é, pois mais *synsets* são possíveis para desambiguá-la.

Figura 4.10: Distribuição de possíveis *synsets* ordenados por tradução

Uma completa especificação das palavras e respectivas quantidades de tradução e *synsets* é apresentada no Apêndice B. A seguir, no Quadro 4.2, são apresentados alguns exemplos de palavras com respectivas quantidade de possíveis *synsets*, das quais algumas encontram-se entre as palavras que apresentaram discordância total entre os anotadores (Tabela 4.1).

A fim de verificar a relação dos dados anotados no *corpus* com a Lei de Zipf (1949), que sugere que palavras mais frequentes tendem a assumir menos significados, analisou-se a distribuição de *synsets* por palavra ordenadas pela frequências no *corpus*, cuja distribuição é ilustrada na Figura 4.11. É perceptível que a curva no gráfico oscila bastante e não se identifica com a curva de Zipf, que é representada por uma parábola.

Ressalta-se que a anotação não se deu em todas as palavras do *corpus*, restringiu-se

Quadro 4.2: Exemplo de Palavras com Respectivas Possibilidades de Desambiguação

partida	23	local	23	título	23
mandato	24	caso	24	energia	25
ataque	25	ajuda	28	passagem	29
cara	29	centro	29	base	30
força	34	ponto	37	posto	38
trecho	38	fila	40	jogo	41
diretor	43	área	54	cargo	59

Figura 4.11: Distribuição de Possíveis *Synsets* Ordenados por Frequência

a somente 10% das palavras mais frequentes em cada coleção de documentos. Com esta restrição, algumas palavras foram anotadas em uma coleção e em outras não. Assim, acredita-se que por estas características da tarefa, a curva apresentada na Figura 4.11 não se identifica com a curva de Zipf. Contudo, como apresentado na Figura 4.7, e como também apresentado pelo princípio do mínimo esforço, as palavras tendem a assumir apenas um significado.

Métodos de DLS Investigados

Neste capítulo, serão abordados os métodos de DLS desenvolvidos neste trabalho, bem como a avaliação dos mesmos. Na Seção 5.1, são apresentadas as características técnicas compartilhadas por todos os métodos de DLS desenvolvidos neste trabalho, que englobam: etapa de pré-processamento de dados; ferramentas externas necessárias; e o processo de recuperação de *synsets*. Na Seção 5.2, é descrito o método heurístico utilizado para aferir valores de *baseline*. Nas Seções 5.3, 5.4 e 5.5, disserta-se sobre os métodos desenvolvidos sendo, respectivamente, uma adaptação do método proposto por Lesk (1986), adaptação sobre o trabalho de Mihalcea and Moldovan (1999) e, por fim, um método baseado em grafo de co-ocorrência para tomar vantagem do cenário multidocumento por meio do algoritmo de Lesk (1986).

Neste capítulo, para melhor entendimento dos métodos de DLS desenvolvidos, serão utilizados exemplos. A fim de padronizar e, conseqüentemente, contribuir para a interpretação das distinções dos métodos apresentados, será empregada apenas uma sentença (doravante, Sentença Exemplo) do corpus CSTNews, da coleção de número 3, que corresponde a “A falha no reversor –mecanismo que ajuda o avião a frear foi ...”.

Os exemplos utilizados serão dados por meio da desambiguação da palavra alvo “avião”. Por tanto, frequentemente, atributos desta palavra serão citados, tais como: possíveis traduções e possíveis *synsets*. Para facilitar a leitura deste capítulo, a listagem destes atributos é apresentada no Quadro 5.1.

Quadro 5.1: Sentença Exemplo e Atributos da Palavra “avião”

Sentença:	“A falha no reversor –mecanismo que ajuda o avião a frear foi ...”
Possíveis traduções:	<i>plane, airplane</i>
Possíveis synsets	
<i>Synset:</i>	<i>airplane, aeroplane, plane</i>
<i>Glosa:</i>	<i>an aircraft that has a fixed wing and is powered by propellers or jets</i>
<i>Exemplos:</i>	<i>the flight was delayed due to trouble with the airplane</i>
<i>Synset:</i>	<i>plane, sheet</i>
<i>Glosa:</i>	<i>((mathematics) an unbounded two-dimensional shape)</i>
<i>Exemplos:</i>	<i>- we will refer to the plane of the graph as the X-Y plane - any line joining two points on a plane lies wholly on that plane</i>
<i>Synset:</i>	<i>plane</i>
<i>Glosa:</i>	<i>a level of existence or development</i>
<i>Exemplo:</i>	<i>he lived on a worldly plane</i>
<i>Synset:</i>	<i>plane, planer, planing machine</i>
<i>Glosa:</i>	<i>a power tool for smoothing or shaping wood</i>
<i>Exemplo:</i>	–
<i>Synset:</i>	<i>plane, carpenter’s plane, woodworking plane</i>
<i>Glosa:</i>	<i>a carpenter’s hand tool with an adjustable blade for smoothing or shaping wood</i>
<i>Exemplo:</i>	<i>the cabinetmaker used a plane for the finish work</i>

5.1 Pré-processamento e Recuperação de *Synsets*

A etapa de pré-processamento, compartilhada pelos métodos desenvolvidos, corresponde à sequência de operações: 1) tokenização dos textos seguindo o padrão requisitado pela ferramenta MXPOST (Ratnaparkhi, 1986); 2) aplicação do etiquetador morfossintático MXPOST (Ratnaparkhi, 1986); 3) lematização das palavras; e 4) remoção de *stopwords*. Após estas etapas, os métodos que serão apresentados possuem comportamentos distintos.

Para recuperação dos possíveis *synsets* de uma palavra, realiza-se uma busca na Wn-Pr em dois estágios, necessários devido à diferença de idioma. Por meio do dicionário bilíngue WordReference®, recuperam-se todas as possíveis traduções da palavra alvo e, somente após isso, retornam-se todos os *synsets* acionados por estas traduções (sem repetições).

Por exemplo, para a palavra “avião”, são retornadas as seguintes traduções “*plane*” e “*airplane*”, que, respectivamente, acionam seis e um *synsets* da Wn-Pr (não são conside-

rados *synsets* que não estejam associados a substantivos). Porém, um dos *synsets* retornados é compartilhado por ambas as traduções. Consequentemente, a palavra “avião”, no processo de recuperação de sentidos empregado nestes trabalhos, possui cinco sentidos diferentes, como pode ser observado no Quadro 5.1.

Nesse processo, pode-se notar que a quantidade de sentidos retornados é influenciada diretamente pela qualidade do dicionário bilíngue, o que impacta diretamente na complexidade e viabilidade de desambiguação. Por exemplo, visto a possibilidade de diversas traduções para uma palavra, a quantidade de *synsets* retornados pode ser grande e, consequentemente, tornar o processo de desambiguação mais complexo, visto a diversidade de sentidos possíveis. Por outro lado, traduções podem não ser listadas para uma determinada palavra e, portanto, para esta palavra específica, não se torna possível a desambiguação.

5.2 Método Heurístico

Assim como muitos trabalhos na literatura e edições de eventos da área, utilizam-se algoritmos mais simples para aferir valores de *baselines* para avaliação de métodos. Adotou-se o método heurístico baseado no sentido mais frequente presente na Wn-Pr. Especificamente neste trabalho, assim como abordado na seção anterior, para recuperação dos possíveis *synsets* de uma palavra, tem-se um processo em duas etapas (tradução e busca na Wn-Pr). Portanto, a recuperação do sentido majoritário dá-se pela utilização da primeira opção de tradução e *synsets* retornados, visto que os resultados obtidos, respectivamente, pelo WordReference® e Wn-Pr são ordenados pela frequência.

Por exemplo, utilizando a exemplificação da seção anterior, a palavra “avião” possui, nesta ordem, as traduções “*plane*” e “*airplane*”. No método heurístico desenvolvido, portanto, usando a tradução “*plane*”, o sentido adotado é constituído pelo conjunto de sinônimos {*airplane*, *aeroplane*, *plane*} e composto pela glosa “*an aircraft that has a fixed wing and is powered by propellers or jets*”.

5.3 Método de Lesk

A adaptação do método de Lesk (1986), pautou-se no trabalho de Banerjee (2002), que apresenta a alteração do método de Lesk para utilização da Wn-Pr como RS, em vez de dicionários.

O método adaptado neste trabalho é composto pelas seguintes etapas: 1) pré-processamento, apresentado na Seção 5.1; 2) recuperação de todos os possíveis *synsets* e traduções das palavras; 3) composição dos contextos de cada palavra alvo, compostos por duas

Tabela 5.1: Análise da Variedade de Rótulos da Adaptação do Algoritmo de Lesk (1986)

–		tradução	glosa	exemplos	glosa+exemplos
Média	Palavras	1,75	42,20	45,47	83,72
	Synset	–	50,29	57,90	103,25
Máximo	Palavras	8	324	444	708
	Synset	–	275	292	523
Mediana	Palavras	2	28,5	21	51
	Synset	–	35	33	67

palavras à esquerda e à direita da PA; 4) construção dos rótulos das palavra do contexto, cujo processo será descrito a seguir; 5) para cada *synset* da PA, computa-se o número de sobreposições de palavras desses e os respectivos rótulos do contexto para determinar o sentido mais adequado.

Banerjee (2002), além de adaptar o algoritmo de Lesk, relata que, devido às distintas composições das informações presentes nos *synsets*, variações no algoritmo de cálculo de similaridade e criação de rótulos são necessárias. Essas variações referem-se a formas de utilização de diferentes dados, como: conjunto de sinônimos, glosa e exemplos. Isto se torna pertinente, pois, por exemplo, alguns *synsets* possuem exemplos e outros não.

Neste trabalho, além das características supracitadas, utilizaram-se também as traduções das palavras, necessárias para recuperações dos *synsets*. Portanto, para formação dos rótulos, compostos por um conjunto de palavras (forma lematizada) e com remoção de *stopwords*, utilizaram-se os seguintes características: (R1) glosas dos *synsets*; (R2) exemplos de uso dos *synsets*; e (R3) possíveis traduções das palavras.

A utilização das possíveis traduções de uma palavra para construção dos respectivos rótulos foi baseada no trabalho de Kilgariff et al. (2000), que apresentam uma simplificação do algoritmo de Lesk (1986), a fim de diminuir a complexidade computacional do algoritmo. Neste método de DLS, o cômputo de similaridade é dado por meio das informações dos possíveis sentidos com as palavras no contexto, não mais por meio de rótulos atribuídos a estas. Neste trabalho, porém, ainda é necessária a atribuição de rótulos, pois algumas palavras possuem diversas traduções. Ressalta-se que rótulos desta abordagem são menores e, conseqüentemente, seguindo o objetivo de Kilgariff et al. (2000), reduz-se, significativamente, a complexidade do algoritmo.

A fim de analisar a composição dos diferentes rótulos, bem como justificar a menor complexidade computacional atingida por meio da composição de rótulos supracitada, realizou-se uma verificação do tamanho dos rótulos concedidos às palavras anotadas no cópús, bem como as características dos *synsets*. Os dados obtidos são apresentados na Tabela 5.1, na qual se pode observar os valores de Média, Máximo e Mediana para cada tipo de rótulo (Tradução, Glosa, Exemplos, Glosa + Exemplos).

É perceptível que rótulos compostos somente pelas possibilidades de tradução são bem menores e, conseqüentemente, requerem menor complexidade computacional. Por exemplo, em média, o tamanho de rótulos com traduções é igual a 1,74 e para glosa é de 42,2. Assim, utilizando-se as informações das glosas dos *synsets* (em média, 50) com os rótulos supracitados, são necessárias, respectivamente, 87 e 2110 comparações entre palavras para o algoritmo de cálculo de similaridade entre os *synsets* e o contexto.

Mesmo reconhecendo a menor complexidade computacional ao empregar rótulos por meio das possíveis traduções, avaliaram-se outras composições destes, a fim de investigar a melhor configuração para o algoritmo. Para tanto, foram adotadas seis estruturas do método: (G-T) compara a glosa do *synset* com rótulos em R3; (S-T) compara exemplos do *synset* com rótulos de R3; (GS-T) compara a glosa e os exemplos do *synset* com rótulos em R3; (G-G) compara a glosa do *synset* com rótulos em R1; (S-S) compara exemplos do *synset* com rótulos em R2; e (GS2) compara glosa e exemplos do *synset* com rótulos em R1 e R2.

Formalmente, seja $s^1, s^2, \dots, s^n = S$ o conjunto de possíveis *synsets* e $p^1, p^2, \dots, p^m = C$ o conjunto das palavras no contexto da palavra alvo e f_r e f_p duas funções de mapeamento, que, respectivamente, mapeiam *synsets* e palavras para um determinado rótulo. Um procedimento, $overlap(r^1, r^2)$, calcula a quantidade de palavras comuns entre dois rótulos. Tem-se que, para cada *synset*, a sua respectiva pontuação ($|s|$) é dada por:

$$\forall s \in S : |s| = \sum_{i=1}^m overlap(f_r(s), f_p(p^i)) \quad (5.1)$$

Por exemplo, para a sentença exemplo, bem como para todas as variações supracitadas, o contexto atribuído à palavra “avião” é composto pelas palavras: “reversor”, “ajuda”, “frear” e “ir”. Assim, utilizando a variação G-T, tem-se que os rótulos das palavras no contexto são: reversor: —; ajuda: *help, aid, support, relief*; frear: *brake*; e ir: *go*.

É importante ressaltar que, neste exemplo, não é listada alguma tradução para a palavra “reversor”, pois utilizou-se a resposta do WordReference® para esboçar a execução automática do método. Após a compilação dos rótulos das palavras, calcula-se a quantidade de palavras comuns entre a glosa dos possíveis *synsets* e estes rótulos, adotando o sentido com maior pontuação.

5.4 Método Irrestrito de Mihalcea

A adaptação do método de [Mihalcea and Moldovan \(1999\)](#) é composta pelas seguintes etapas: 1) etapa de pré-processamento detalhada na Seção 5.1; 2) recuperação das traduções e, posteriormente, *synsets*; 3) agrupamento dos pares de palavras (palavra alvo e

de contexto); 4) construção das *queries* de busca; e 5) realização das buscas por meio da API do buscador Bing® para cômputo dos resultados, para determinar o sentido correto.

Ressalta-se que, diferentemente de [Mihalcea and Moldovan \(1999\)](#), que agrupam os pares de palavras alterando as classes morfossintáticas (substantivo-substantivo, substantivo-verbo, etc.), neste trabalho, utilizaram-se apenas pares de palavras formados por um substantivo e um verbo. Esta opção foi empregada porque os autores apresentam que esta foi a combinação com melhor desempenho.

Por exemplo, na sentença exemplo, o par para a palavra “avião” é “frear”, visto que este é o verbo mais próximo da palavra alvo. Neste cenário, portanto, as *queries* são constituídas com a tradução da palavra “frear (*brake*)” e os conjuntos de sinônimos dos *synsets*, seis no total, da palavra “avião”, como se segue:

1. Synset: *airplane, aeroplane, plane*
Query: “airplane brake” OR “aeroplane brake” OR “plane brake”
2. Synset: *plane, sheet*
Query: “plane brake” OR “sheet brake”
3. Synset: *plane*
Query: “plane brake”
4. Synset: *plane, planer, planing machine*
Query: “plane brake” OR “planer brake” OR “planing machine brake”
5. Synset: *plane, carpenter’s plane, woodworking plane*
Query: “plane brake” OR “carpenter’s plane brake” OR “woodworking plane brake”

Outra alteração empregada foi na criação das *queries*. No trabalho dos autores supracitados, são utilizados os conjuntos de sinônimos dos *synsets* e a palavra de contexto. Neste trabalho, devido à desambiguação de palavras em PT-BR por meio da Wn-Pr, as chaves de busca são constituídas com os conjuntos de sinônimos dos *synsets* e todas as possíveis traduções da palavra de contexto. Outra diferenciação é a ferramenta de busca utilizada, o Bing®, em vez do uso do Altavista®, que é adotado por [Mihalcea and Moldovan \(1999\)](#).

Para exemplificar este método, adota-se a sentença exemplo e a sequência de *queries* enumeradas anteriormente. Após execução do algoritmo, têm-se as seguintes pontuações (quantidade de páginas retornadas por cada chave de busca): 1) 3990; 2) 7340; 3) 549; 4) 3170 e 5) 561. Assim, o *synset* adotado para a palavra alvo é sua segunda opção, pois este apresentou maior quantidade de páginas retornadas. Assim, para este caso, o método de DSL apresentado não acertou o sentido correto da palavra.

5.5 Desambiguação com Grafo de Co-ocorrência Multidocumento

O cenário multidocumento, inserido em diversas aplicações do PLN, apresenta uma maior quantidade de informação, que, em geral, métodos habituais de DLS não usufruem. Partindo da heurística que diz que uma palavra tende a manter um mesmo sentido em um discurso (Mihalcea, 2006), evidenciada via análise do *cópus* CSTNews descrita no Capítulo 4, o método de desambiguação apresentado nesta seção admite que uma palavra possui apenas um sentido na coleção de textos.

O método consiste na aplicação do algoritmo apresentado na Seção 5.3, mais especificamente a variação G-T, porém, com alteração no contexto da palavra alvo. A hipótese deste método é que, admitindo que uma palavra alvo P possua apenas um sentido na coleção, determinar seu significado por meio das palavras que mais co-ocorreram implica a melhoria da desambiguação. Ou seja, em vez de usar as palavras em torno de P e, possivelmente, adotar vários sentidos para esta, computam-se as palavras que mais ocorreram próximas a P em toda coleção e, após isto, determina-se o sentido mais adequado.

Para tanto, um grafo $G(V,A)$ de co-ocorrência de uma coleção de textos C é construído durante a etapa de pré-processamento. G é constituído por um conjunto de vértices V , que representa o vocabulário empregado em C , e um conjunto de arestas A , que representa as co-ocorrências entre palavras (vértices) nos textos. Ressalta-se que não há repetição de palavras no grafo, ou seja, cada vértice referencia, exclusivamente, uma única palavra em C .

Assim, todo $v \in V$ representa uma palavra $p \in C$, referenciada por sua forma lematizada, que, por sua vez, possui apenas um único vértice; toda $a(v', v'') \in A$ relaciona dois vértices (v', v'') diferentes e representa a co-ocorrência entre duas palavras $p', p'' \in C$ (referenciadas pelos vértices); e toda $a \in A$ possui peso w , que representa a quantidade de vezes que duas palavras (vértices) co-ocorreram na coleção. Neste trabalho, duas palavras p' e p'' co-ocorrem se e somente se estas estão presentes dentro de uma mesma janela de tamanho N .

Para construir G , dada uma coleção de textos C , têm-se as seguintes etapas: 1) etapa de pré-processamento (apresentada na Seção 5.1); 2) processo de criação de vértices utilizando uma janela de palavras com tamanho n , sendo que para cada palavra presente nesta, adiciona-se, caso ainda não exista, um vértice que a referencie em G ; 3) para cada par de palavras (p', p'') na janela, caso não exista, cria-se uma aresta com peso 1 entre seus respectivos vértices em G e, caso tal aresta exista, é adicionado um em seu peso atual; 4) após estas etapas, a janela é deslizada em uma palavra à direita; e 5) este processo se repete até o final de todos os textos.

Na Figura 5.1, é apresentado um exemplo de grafo de co-ocorrência. Nesta Figura,

para simplificar, é utilizada uma janela de tamanho 3 e a sequência A, B, C, D, E, A, D. Neste exemplo, tem-se as seguintes sequências de janelas: ABC; BCD; CDE; **DEA**; e **EAD**. Nota-se que A e E co-ocorrem duas vezes com D, visto que estes estão presentes em duas janelas distintas. Portanto, as arestas A-D e D-E possuem peso 2. Entretanto, é importante salientar que, apesar de E e A também co-ocorrerem em duas janelas, estas ocorrências, demarcadas em negrito, são as mesmas instâncias, ou seja, apesar de estarem em janelas diferentes, são as mesmas palavras e, portanto, não contabilizam duas co-ocorrências.

Figura 5.1: Exemplo de Grafo de Co-ocorrência

Após a criação do grafo de co-ocorrência, para cada palavra a ser desambiguada, usando-se os pesos das arestas, cria-se uma lista das m palavras que mais co-ocorreram (em caso de empate entre duas ou mais palavras, é utilizado um sorteio). Com as listas compiladas para cada palavra alvo, aplica-se o algoritmo apresentado na Seção 5.3, utilizando-se estas listas como contexto das palavras alvo.

Na Figura 5.2, é apresentado o grafo de co-ocorrência, construído por meio de uma janela de tamanho 3, para desambiguação da palavra “avião”, presente na sentença exemplo. Neste grafo, pode-se observar todas as palavras que co-ocorrem com a PA em toda a coleção. Para simplificação, apenas as palavras que co-ocorrem com a “avião” foram listadas, e apenas as arestas com pesos maiores que 1 foram ponderadas, ou seja, todas as arestas que não possuem peso indicados são ponderadas com valor igual a 1.

Utilizando este grafo de co-ocorrência, tem-se que o contexto da palavra alvo é dado por: “frear” e “ter”. Pode-se observar que, para este método multidocumento, o contexto atribuído à palavra “avião” na sentença exemplo é diferente do método apresentado na Seção 5.3. Após a listagem do contexto da PA, aplica-se o algoritmo G-T para desambiguação, no qual computa-se a quantidade de palavras sobrepostas entre as glosas dos

synsets e as traduções das palavras no contexto. Neste cenário, o sentido atribuído à PA por meio deste algoritmo é o primeiro *synset* apresentado no Quadro 5.1, que corresponde ao significado anotado no *corpus*, assim, portanto, o algoritmo acertou neste caso.

Figura 5.2: Exemplo de Grafo de Co-ocorrência para Palavra “avião” por meio do Cluster de número 3

5.6 Avaliação

O conjunto de testes empregado para avaliação dos métodos propostos foram os exemplos manualmente anotados no *corpus* CSTNews. Para comparar o desempenho dos métodos, foi utilizado o método heurístico que aplica o sentido mais frequente (descrito na Seção 5.2) para aferir valores de *baseline*.

A avaliação ocorreu sobre duas tarefas, *all-words* e *lexical-sample*. A primeira foi voltada à avaliação geral dos métodos em todo o *corpus*, sendo necessária a desambiguação de todas as palavras anotadas. A segunda tarefa foi empregada para avaliar o comportamento dos métodos em palavras mais ambíguas (que foram anotadas com mais de 2 sentidos diferentes), visto que a maioria das palavras apresentaram apenas um sentido (vide Seção 4.1), normalmente o sentido mais frequente, e acabam privilegiando o método heurístico.

Na Tabela 5.2, a avaliação da tarefa *all-words* é apresentada. Nesta tabela, todos os métodos propostos são listados. As métricas utilizadas são (apresentadas na Seção 2.3): P) precisão; A) abrangência; C) cobertura; e Ac) acurácia. Ressalta-se que, para o método heurístico, o valor aferido pela acurácia é igual ao respectivo valor de precisão, portanto, não é apresentada para este método.

É importante ressaltar que algumas palavras do *corpus*, que deveriam ter sido desam-

Tabela 5.2: Avaliação *all-words*

Método	P(%)	C(%)	A(%)	Ac(%)
H	51.00	51.00	100	–
Lesk (G-T)	42.20	41.20	91.10	41.20
Lesk (S-T)	42.20	41.10	91.10	41.10
Lesk (GS-T)	41.80	38.00	91.10	38.00
Lesk (G-G)	27.00	24.60	91.10	24.60
Lesk (S-S)	28.20	25.70	91.10	25.70
Lesk (GS2)	26.80	24.40	91.10	24.40
Mihalcea	39.71	39.47	99.41	39.59
G3	49.56	43.90	88.59	43.90
G5	46.87	41.80	87.65	41.80

biguadas pelos métodos, não apresentaram, de forma automática, nenhuma possibilidade de desambiguação, ou seja, o método de recuperação de *synsets* empregado neste trabalho não retornou nenhum resultado (seja por impossibilidade de tradução automática da PA, ou que as possíveis traduções da PA não acionaram nenhum *synset*). Assim, tendo em vista que essa característica é comum a todos os algoritmos de DLS desenvolvidos neste trabalho, esses casos foram desconsiderados para o cálculo do valor de abrangência. Assim, decréscimos no valor de abrangência dos métodos representam deficiências na etapa de pré-processamento e/ou na representação do contexto da PA.

Pela primeira avaliação, pode-se observar que os cinco melhores métodos desenvolvidos, exceto o método heurístico, são em ordem: 1) G3, método com grafo multidocumento com janela de tamanho 3; 2) G5, método com grafo multidocumento com janela de tamanho 5 ; 3) G-T e S-T; 4) GS-T; e 5) Mihalcea. Nenhum método desenvolvido foi superior ao método *baseline*, contudo, ressalta-se que, por meio do Teste t de *Student* com 95% de confiança, a diferença de precisão do algoritmo G3 e o método Heurístico (H) é estatisticamente irrelevante.

Para a segunda avaliação, a tarefa *lexical sample*, foram utilizadas apenas 20 palavras, somente aquelas que apresentaram mais de dois sentidos anotados no corpus. Utilizou-se apenas a métrica de Precisão para esta avaliação, visto que, nesta tarefa, o objetivo é analisar o percentual de acerto dos métodos sobre um conjunto bem definido, ou seja, o desempenho dos métodos sobre as palavras selecionadas.

Na Tabela 5.3, são apresentados os resultados obtidos. Nas linhas, são dispostas as palavras em ordem alfabética e, nas colunas, os métodos desenvolvidos. Assim, o valor apresentando na célula de linha l e coluna c corresponde à precisão do método da coluna c para a palavra na linha l . Por exemplo, o método G-T, coluna 3, apresentou valor de precisão igual a 41,70% para a palavra “agência”, na linha 3.

Os métodos desenvolvidos são referenciados por siglas da seguinte forma: MH) Método

Tabela 5.3: Avaliação *lexical-sample*

Palavra	H	G-T	S-T	GS-T	G-G	S-S	GS2	M	GC(3,5)
acordo	12.50	6.30	12.50	6.30	18.80	6.30	6.30	13.33	12.50
agência	41.70	41.70	41.70	41.70	0.00	0.00	0.00	20.00	25.00
ano	90.50	69.60	86.30	67.60	0.00	22.50	39.20	94.83	47.22
área	16.70	5.60	5.60	5.60	11.10	0.00	5.60	0.00	5.56
centro	61.50	0.00	80.00	0.00	20.00	70.00	30.00	57.14	36.36
competição	0.00	6.70	0.00	0.00	46.70	0.00	0.00	35.71	0.00
estado	33.30	33.30	16.70	16.70	41.70	0.00	8.30	28.57	30.00
filho	25.00	25.00	25.00	25.00	0.00	25.00	0.00	25.00	25.00
hora	50.00	50.00	50.00	50.00	50.00	0.00	25.00	50.00	0.00
investigação	74.10	61.50	73.10	61.50	38.50	73.10	69.20	60.00	18.75
local	30.00	0.00	30.00	0.00	0.00	0.00	0.00	33.33	17.65
obra	42.50	23.10	0.00	0.00	0.00	0.00	0.00	64.71	26.83
ouro	0.00	0.00	0.00	0.00	30.00	0.00	0.00	0.00	0.00
país	39.20	38.00	40.00	38.00	4.00	34.00	32.00	76.47	17.46
parte	10.00	0.00	10.00	10.00	0.00	10.00	0.00	14.77	10.00
partida	38.50	27.80	16.70	16.70	5.60	0.00	0.00	12.50	52.63
presidente	16.40	19.00	15.90	15.90	6.30	4.80	0.00	14.46	9.41
resultado	70.00	65.00	55.00	55.00	0.00	55.00	65.00	78.95	66.67
tempo	0.00	14.30	28.60	35.70	7.10	0.00	7.10	0.00	0.00
vez	0.00	0.00	10.50	10.50	0.00	0.00	0.00	0.00	0.00
vôo	3.60	0.00	0.00	0.00	0.00	4.80	4.80	0.00	0.00
Total > H	-	3	4	2	5	2	2	8	1
Total = H	-	6	8	6	2	6	3	5	7
Total >= H	-	9	12	8	7	8	5	13	8
Total < H	-	12	9	13	14	13	16	8	13
Média	27.88	23.19	28.46	21.72	13.32	14.55	13.93	32.37	19.10

heurístico; G-T) Método de Lesk com a variação GT; ST) Método de Lesk com a variação ST; GS-T) Método de Lesk com variação o GS-T; G-G) Método de Lesk com variação G-G; S-S) Método de Lesk com variação S-S; GS2) Método de Lesk com variação GS2; M) Método de Michalcea; GC) Método baseado em grafo de co-ocorrência. Neste experimento, a variação do método GC com janelas de tamanho 3 e 5 não apresenta valores diferentes, portanto, estes são apresentados na mesma coluna.

Por exemplo, na segunda linha, tem-se a avaliação dos métodos desenvolvidos sobre a palavra “agência”. Pode-se observar que três variações do método de lesk (G-T, S-T e GS-T) obtiveram precisão igual ao valor *baseline*, 41.70. Entretanto, os demais métodos não acertaram nenhuma instância da palavra.

Nas últimas linhas da Tabela 5.3, são apresentadas as quantidades de vezes em que os métodos desenvolvidos obtiveram número de precisão de desambiguação maior, igual, maior ou igual e menor ao número de desambiguações corretas do método heurístico. Por fim, também é apresentada a média da precisão dos respectivos métodos. Por exemplo, o método da Mihalcea apresentou precisão maior ou igual ao método heurístico em um total de 13 vezes. Este algoritmo também obteve a maior média de precisão, 32,37.

Pode-se observar que o método baseado em Grafo, que obteve melhor desempenho

na tarefa de *all-words*, mostra-se pouco eficiente neste cenário. Somente para algumas palavras (“acordo”, “filho”, “parte”, “partida”), a precisão aferida é superior ou igual ao valor *baseline*. Isso se explica pela principal restrição do método, que atribui somente um sentido por palavra em uma mesma coleção. Ou seja, para palavras mais ambíguas, que, normalmente, apresentam sentidos distintos em uma mesma coleção, a aplicação deste método é pouco eficiente.

Neste contexto, é importante ressaltar o caso da palavra “partida”, que foi um exemplo onde o método baseado em grafo apresentou a maior precisão, 52,63. Esta palavra ocorreu e foi anotada em três coleções do *corpus* com três *synsets* diferentes. Em uma destas coleções, a palavra “partida” foi anotada com um mesmo sentido. Assim, sendo que o método G3 e G5 atribuem somente um sentido por palavra em uma coleção, e o método acertou este *synset*, a precisão destes métodos mostrou-se superior neste caso.

Dentre as variações empregadas para o algoritmo de Lesk (1986), pode-se observar que, quanto maior a quantidade de informação presente nos rótulos, pior o desempenho dos métodos. É perceptível que as variações G-T e S-T, que empregam a simplificação do algoritmo original de Lesk, proposto por Kilgariff et al. (2000), são superiores às demais configurações tanto na tarefa *all-words* quanto na tarefa *lexical-sample*. Por exemplo, na segunda tarefa, a variação G-G do algoritmo, em relação ao *baseline*, foi superior em 5 casos, enquanto que as variações S-T e G-T, respectivamente, foram melhores em 4 e 3 testes. Contudo, em uma visão mais abrangente, a julgar pelo número de vezes que estes algoritmos obtiveram resultados melhores ou iguais ao *baseline*, o método G-G foi inferior.

A adaptação do método de Mihalcea and Moldovan (1999) mostrou-se inferior na tarefa de *all-words*, sendo superado por outros quatro métodos. Entretanto, na segunda avaliação, obteve os melhores resultados, sendo superior ao método heurístico em 8 casos, inferior também em 8 situações e melhor ou igual para 13 palavras. Do ponto de vista numérico de comparação, considerando apenas a parte inferior da Tabela 5.3, esse algoritmo é equivalente ao método heurístico. Contudo, é importante ressaltar que a média da precisão obtida por esse método, 32,37, é superior aos demais, inclusive à média do *baseline*. O melhor desempenho deste método, especificamente nesta tarefa, se justifica por palavras fortemente ambíguas apresentarem mais possibilidades de *synsets*, que, consequentemente, contribuem para a construção das *queries* necessárias a esse algoritmo.

A seguir, na Figura 5.3, é apresentada uma representação de relações entre conjuntos de palavras desambiguadas pelos principais métodos desenvolvidos neste trabalho, a adaptação G-T do algoritmo de Lesk (1986), o método baseado em grafo multidocumento com janela de tamanho 3 (G3) e a adaptação do método de Mihalcea and Moldovan (1999). Nesta representação, cada método é identificado por um conjunto (graficamente retratado por uma circunferência), que representa as palavras que foram corretamente

desambiguadas pelo algoritmo. Assim, as regiões de intersecção entre os conjuntos identificam a quantidade de palavras corretamente desambiguadas por dois ou três métodos, as regiões isoladas representam o número de palavras acertadamente desambiguadas por somente um método. Por fim, na região externa a todas as circunferências, é mostrada a quantidade de palavras que não foram corretamente desambiguadas por nenhum dos métodos.

Figura 5.3: Relação de Palavras Desambiguadas pelos Métodos

Nesta representação, pode-se perceber que o método adaptado de [Mihalcea and Moldovan \(1999\)](#), de forma isolada, desconsiderando as palavras compartilhadas pelos métodos, distinguiu-se com maior força dos demais métodos, pois desambiguou um número maior de palavras que não foram compartilhadas pelos outros algoritmos. Este fato indica um viés do algoritmo, ou seja, o algoritmo tende a acertar palavras de uma determinada característica, que, neste caso, foram palavras que possuíam um número maior de *synsets* ou que o contexto, representado por apenas um verbo, possuía mais *synsets*, contribuindo diretamente na construção de *queries* de busca, que é a característica principal deste algoritmo.

Analisando o algoritmo G3 e GT, desconsiderando as palavras que ambos acertaram, tem-se, respectivamente, 161 ($75+87$) e 100 ($62+38$) palavras desambiguadas corretamente. Tendo em vista que o método G3 usa o algoritmo GT no processo de desambiguação, porém, utilizando o cenário multidocumento por meio de um grafo de co-ocorrência de palavras para representação do contexto, o melhor desempenho do algoritmo G3 evidencia que a utilização do cenário multidocumento é promissora para a DLS, mesmo adotando apenas um sentido por palavra em uma determinada coleção de textos.

Com esta avaliação dos métodos desenvolvidos, em relação às hipóteses deste trabalho, pode-se concluir que:

- É possível desenvolver métodos de DLS para o PT-Br utilizando a Wn-Pr, de tal forma que sejam superiores ao método *baseline*: Do ponto de vista geral, tarefa

all words, os métodos desenvolvidos mostraram-se inferiores ao método heurísticos. Contudo, é importante salientar que a maioria das palavras anotadas no corpus apresentou somente um *synset* anotado, e, normalmente, foi o significado mais frequente, o que representa um cenário promissor para o método heurístico. Além disso, nesta tarefa, a diferença de desempenho entre o método *baseline* e o G3 é estatisticamente irrelevante. Do ponto de vista da tarefa *lexical sample*, alguns métodos desenvolvidos apresentaram-se superiores para diversas palavras.

- Utilizar informações do cenário multidocumento pode contribuir para o processo de DLS: Esta hipótese foi comprovada, pois os métodos multidocumento, G3 e G5, mostraram-se superiores aos demais métodos, especificamente no tarefa de *all words*. É importante salientar que estes métodos foram superiores devido ao cenário do corpus, onde a maioria das palavras tende a manter um mesmo significado em uma mesma coleção. Devido ao tamanho do corpus, não se pode afirmar que outros cenários multidocumento também apresentem esta característica.

Conclusões

Neste Capítulo, serão apresentadas as conclusões e discussões finais deste trabalho de mestrado, bem como as contribuições as contribuições. Para tanto, a fim de organização, as conclusões deste trabalho serão divididas em duas partes. Na primeira parte, Seção 6.1, será dissertado sobre as conclusões em torno da anotação do CSTNews. Já na segunda parte, Seção 6.2, serão apresentas as conclusões sobre os métodos desenvolvidos.

Na Seção 6.3, serão apresentadas as contribuições, tanto a respeito do córpus anotado quanto aos métodos. Por fim, na Seção 6.4, serão apresentadas as sugestões de trabalho futuros.

6.1 Conclusão sobre Anotação do Córpus

Neste trabalho, para subsidiar as etapas de desenvolvimento e avaliação dos métodos de DLS, bem como a verificação de algumas hipóteses, anotou-se o córpus CSTNews, descrito na Seção 4.1.

O valor de concordância obtido, segundo a medida kappa, foi de 72.9, o que representa um bom nível de concordância entre os anotadores e, consequentemente, dá indícios de que a tarefa pode ser automatizada. A fim de comparação, para o inglês, Mihalcea (2007) apresenta concordância de 87.1%. Entretanto, ressalta-se que a tarefa de anotação deste trabalho possui complexidades maiores devido à utilização de um RS destinado a outra língua, como necessidade de tradução e existência de lacunas lexicais.

É importante ressaltar que a tarefa de anotação realizada neste trabalho apresenta características que dificultam o processo, tais como: utilização de um RS do inglês para

desambiguação de palavras no PT-BR e presença do cenário multidocumento. Acredita-se que a utilização de repositórios semelhantes à estruturação das *wordnets* para o PT-BR contribuiria para um maior valor de concordância.

O corpus apresenta diversos exemplos de palavras que foram anotadas com apenas um *synset* em todo o corpus. Na minoria dos casos, foram anotadas palavras que obtiveram no máximo cinco sentidos em todo o corpus. Contudo, algumas palavras possuem diversas instâncias anotadas e, conseqüentemente, podem ser utilizadas em técnicas de AM.

É importante ressaltar que, devido às características multidocumento do CSTNews, experimentos para verificação do comportamento das palavras e respectivos sentidos neste cenário foram desenvolvidos. Essas investigações foram considerados pelos autores como a grande contribuição desse processo de anotação, visto que impulsionaram o desenvolvimento de métodos de DLS para o cenário multidocumento e várias aplicações do PLN engajam-se neste contexto. Com isso, observou-se que as palavras tendem a assumir um sentido por contexto e, mais especificamente neste trabalho, em uma coleção de textos relacionados.

Estes experimentos, além de avaliar o comportamento dos sentidos neste cenário, enfatizam que a utilização de métodos de DLS que determinam apenas um sentido para instâncias de uma palavra em um mesmo discurso é promissora, pelo menos, no domínio de notícias jornalísticas. Contudo, o corpus CSTNews, comparado com outros corpus encontrados na literatura, não é considerado tão grande. Portanto, esta afirmativa é mais um indicativo da utilização de um sentido por discurso.

Para anotar o corpus, foi desenvolvida uma ferramenta intitulada NASP. Esta ferramenta é facilmente re-configurável para que outros cenários sejam empregados, como: anotação de outras classes morfosintáticas (diferentes de substantivos) e outros idiomas (alterando o dicionário utilizado). Portanto, esta ferramenta pode impulsionar diversos trabalhos de DLS que necessitam de etapas de anotação de corpus para um idioma diferente do inglês.

6.2 Conclusão sobre os Métodos de DLS Desenvolvidos

Os métodos de DLS foram avaliados sobre o corpus CSTNews em duas tarefas. A primeira tarefa, *all-words*, foi destinada à verificação geral dos métodos, sendo necessária a desambiguação de todas as palavras anotadas no corpus. Já a segunda tarefa, *lexical-sample*, objetivou a verificação do comportamento dos métodos sobre palavras mais ambíguas no corpus, aquelas que apresentaram mais de dois sentidos diferentes anotados.

Na primeira tarefa, tem-se que o método heurístico, que emprega a tradução e respectivo *synset* mais frequente, apresentou melhores resultados. Contudo, resalta-se que,

observando as distribuições de palavras e respectivos sentidos anotados no *cópus* (Figura 4.7), tem-se que a maioria destas possuem apenas um sentido, que, normalmente, são os mais frequentes. Tal distribuição indica um cenário que favorece o método heurístico.

Dentre as adaptações do método de Lesk, tem-se que as que obtiveram melhores resultados foram aquelas que empregaram menor quantidade de informação para criação de rótulos. De forma mais específica, tratam-se de adaptações do trabalho de Kilgarriff et al. (2000), no qual o *cômputo* para adotar o melhor significado é dado pela quantidade de sobreposições de palavras entre as definições do sentido e as palavras no contexto, não mais em rótulos atribuídos a essas.

Devido à necessidade de tradução e que uma palavra no PT-BR pode assumir inúmeras traduções para o inglês, estas variações utilizam-se de rótulos compostos por suas respectivas traduções. Duas variações do método apresentam o mesmo valor nos experimentos, uma com utilização da Glosa e outra com informações dos Exemplos dos *synsets*.

É importante também salientar que as variações do método de Lesk que empregaram mais informações dos *synset*, tanto para o cálculo de sobreposição de palavras, quanto para atribuição de rótulos às palavras do contexto, não apresentaram bons resultados. Estes resultados, são equivalentes a experimentos relatados por outros autores, como Kilgarriff et al. (2000); Sinha and Mihalcea (2007). Outro fator importante, que indica resultados pouco satisfatórios para uma maior quantidade de informação, é o efeito cascata de erros oriundos de possíveis falhas de tradução.

Observada a variação de sentido por palavra em uma mesma coleção (como apresentado na Seção 4.3), foi desenvolvido um método baseado em grafo que admite apenas um sentido por palavra em um mesmo contexto, neste caso, uma coleção. Este método, que empregou a melhor variação do método de Lesk, obteve melhores resultados em relação aos demais, exceto ao método heurístico. Neste último caso, após verificação de testes estatísticos, obteve-se que a diferença de desempenho entre o método de grafo e o método heurístico é insignificante.

Este método foi avaliado sobre duas configurações, que correspondiam à utilização de diferentes tamanhos de janelas para criação do grafo de co-ocorrência e para desambiguação (vide Seção 5.5). Novamente, assim como em outros trabalhos na literatura, a utilização do menor contexto obteve melhores resultados.

Os melhores resultados alcançados por este método são diretamente relacionados ao cenário apresentado no *cópus*. Entretanto, na segunda tarefa, este método não se demonstrou melhor do que a adaptação do algoritmo de Lesk.

6.3 Contribuições

As contribuições deste trabalho podem ser divididas em teóricas e técnicas. Dentre as contribuições teóricas, tem-se a investigação da desambiguação lexical no cenário multidocumento, que se constitui na experimentação em *corpus* e dos métodos de DLS desenvolvidos e avaliados.

Neste contexto, foram desenvolvidos métodos de DLS para o PT-BR por meio da utilização de recursos do inglês, a Wn-Pr, caracterizando-se como um trabalho de investigação multilíngue. Portanto, a metodologia e os resultados obtidos neste trabalho podem ser utilizados para pesquisas em outras línguas que não possuam recursos linguísticos disponíveis.

Dentre as contribuições técnicas, tem-se a ferramenta de anotação NASP, voltada à desambiguação de palavras em contexto mono e multidocumento. Outro recurso importante, que pode contribuir com pesquisas futuras, é a anotação de desambiguação do *corpus* CSTNews.

6.4 Trabalhos Futuros

Dada a característica multilíngue deste trabalho, pretende-se investigar a utilização de RS de outros idiomas. Por exemplo, desenvolver estes métodos de DLS para o PT-BR usando RS para línguas européias, como espanhol. Por outro lado, pretende-se investigar a metodologia deste trabalho, bem como os métodos de DLS desenvolvidos, em tarefas de desambiguação em outros idiomas, principalmente aqueles que possuem poucos recursos, como árabe.

A fim de investigar com mais afinco os métodos de DLS multidocumento, pretende-se aplicá-los em cenários da língua inglesa, visto que para este idioma existem mais métodos de desambiguação, e comparações mais relevantes com outros métodos encontrados na literatura.

Pretende-se ainda, estender este trabalho para demais classes morfossintáticas, como verbos e adjetivos. Assim, pretende-se primeiramente, ampliar a anotação do *corpus* CSTNews para as demais classes morfossintáticas e, posteriormente, avaliar os métodos de DLS.

Uma outra possibilidade futura para extensão deste trabalho, principalmente no contexto da tarefa *lexical sample*, é investigar métodos de Aprendizado de Máquina usando o *corpus* anotado como conjunto de treinamento. Para tanto, é necessário selecionar algumas palavras anotadas para geração de classificadores, pois nesta abordagem de DLS é necessária a criação de um classificador diferente para cada palavra, e, possivelmente,

estender o *cópus* com mais exemplos anotados, para dar maior volume ao conjunto de treinamento e validação dos classificadores.

Referências Bibliográficas

- Agirre, E. and P. Edmonds (2006). *Word Sense Disambiguation: Algorithms and Applications*, Chapter Introduction, pp. 1–28. Springer.
- Agirre, E. and A. Soroa (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of 12th Conference of the European Chapter of the ACL*, pp. 33–41.
- Agostini, V., R. T. de Camargo, A. Di Felippo, and T. A. S. Pardo (2012). *Alinhamento manual dos sumários humanos e dos textos-fonte do corpus multidocumento CSTNews*. Technical report, Instituto de Ciências da Computação e Matemática Computação.
- Aires, R. V. X. (2000). Implementação, adaptação, combinação e avaliação de etiquetadores para o português do brasil. Master’s thesis, Instituto de Ciências Matemáticas e de Computação.
- Aleixo, P. and T. A. S. Pardo (2008). *CSTNews: Um Corpus de Textos Jornalísticos Anotados segundo a Teoria Discursiva Multidocumento CST (Cross-document Structure Theory)*. Technical Report 326, Instituto de Ciências Matemáticas e de Computação.
- Banerjee, S. (2002). Adapting the lesk algorithm for word sense disambiguation to wordnet. Master’s thesis, Department of Computer Science, University of Minnesota.
- Banerjee, S. and T. Pedersen (2002). An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of 3th International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 136–145.
- Black, E. W. (1988). An experiment in computational discrimination of english word senses. *IBM Journal of Research and Development Vol. 32*, pp. 185–194.
- Bond, F. and K. Paik (2012). A survey of wordnets and their licenses. *Small Vol. 8*, pp. 5.

- Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of 17th International World-Wide Web Conference (WWW 1998)*.
- Cardoso, P. C. F., E. G. Maziero, M. L. R. C. Jorge, E. M. R. Seno, A. Di Felippo, L. H. M. Rino, M. d. G. V. Nunes, and T. A. S. Pardo (2011). CSTNews – a discourse-annotated corpus for single and multi-document summarization of news texts in brazilian portuguese. In *Anais do III Workshop “A RST e os Estudos do Texto”*, Cuiabá, MT, Brasil, pp. 88–105. Sociedade Brasileira de Computação.
- Cardoso, P. C. F., A. P. Rassi, E. G. Maziero, F. A. A. Nóbrega, J. W. C. Souza, M. S. Dias, M. L. R. C. Jorge, P. P. B. Filho, R. T. Camargo, V. Agostini, A. D. Felippo, L. H. M. Rino, and T. A. S. Pardo (2012). *Anotação de subtópicos do córpus multidocumento CSTNews*. Technical report, Instituto de Ciências da Computação e Matemática Computação–ICMC–USP.
- Carletta, J. (1996). Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics Vol. 22*, pp. 249–254.
- Che, W. and T. Liu (2010). Jointly modeling wsd and srl with markov logic. In *Proceedings of 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China, pp. 161–169. Coling 2010 Organizing Committee.
- Di Felippo, A. (2008). *Delimitação e Alinhamento de Conceitos Lexicalizados no Inglês Norte-americano e no Português Brasileiro*. Ph. D. thesis, Faculdade de Ciências e Letras, Universidade Estadual Paulista.
- Dias da Silva, B. C. (2005). A construção da base da wordnet.br: Conquistas e desafios. In *Anais do XXV Congresso da Sociedade Brasileiro de Computação*.
- Domingo, E. and R. E. Roxas (2006). Automatic target word disambiguation using syntactic relationships. In *Proceedings of 20th Pacific Asia Conference on Language, Information and Computation*, pp. 374–377.
- Edmonds, P. (2002). Senseval: The evaluation of word sense disambiguation systems. In *Proceedings of ELRA Newsletter*, Vol. 7.
- Fellbaum, C. (1998). *WordNet An Eletronic Lexical Database*. MIT Press.
- Gantz, J. and D. Reinsel (2011). *Extracting Value from Chaos*. Technical report, IDC iView.
- Hearst, M. A. (1991). Noun homograph disambiguation using local context in large text corpora. In *Proceedings of University of Waterloo*, pp. 1–22.

- Hovy, E., M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel (2006). Ontonotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*, New York, pp. 57–60.
- Ide, N. and J. Véronis (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics Vol. 24*, pp. 2–40.
- Jiang, J. and D. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the Int’l. Conf. on Research in Computational Linguistics*, pp. 19–33.
- Jurafsky, D. and J. H. Martin (2009). *Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (2nd ed.). Prentice Hall. Pearson.
- Kilgariff, A., B. England, and J. Rosenzweig (2000). English senseval: Report and results. In *Proceedings of 2nd International Conference on Language Resources and Evaluation*, pp. 1239–1244.
- Leacock, C. and M. Chodorow (1998). *Combining local context and WordNet similarity for word sense identification*, pp. 305–332. In C. Fellbaum (Ed.), MIT Press.
- Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of 5th Annual International Conference on Systems Documentation*, New York, NY, USA, pp. 24–26. Association for Computing Machinery.
- Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML ’98, San Francisco, CA, USA, pp. 296–304. Morgan Kaufmann Publishers Inc.
- Machado, I. M., R. O. de Alencar, R. de Oliveira Campos Junior, and C. A. Davis (2011). An ontological gazetteer and its application for place name disambiguation in text. *Journal of the Brazilian Computer Society Vol. 17*, pp. 267–279.
- Mann, W. C. and S. A. Thompson (1987). *Rhetorical Structure Theory: A Theory of Text Organization*. Technical report, ISI: Information Sciences Institute.
- Maziero, E. G., T. A. S. Pardo, A. Di Felippo, and B. C. Dias da Silva (2008). A base de dados lexical e a interface web do tep 2.0 — thesaurus eletrônico para o português do Brasil. In *Anais do VI Workshop em Tecnologia da Informação e da Linguagem Humana (TIL)*, pp. 390–392.

- Melli's, G. (2011). Semcor corpus. *online*. Available in: http://www.gabormelli.com/RKB/SemCor_Corpus.
- Mihalcea, R. (1998). Semcor semantically tagged corpus. Unpublished Manuscript.
- Mihalcea, R. (2006). *Word Sense Disambiguation: Algorithms and Applications*, Chapter Knowledge-Based Methods for WSD, pp. 107–131. Springer.
- Mihalcea, R. (2007). Using wikipedia for automatic word sense disambiguation. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York, pp. 196–203. Association for Computational Linguistics.
- Mihalcea, R. and P. Edmonds (2011). Senseval org. Online. Accessed on September 1, 2011.
- Mihalcea, R. and D. I. Moldovan (1999). A method for word sense disambiguation of unrestricted text. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics*, College Park, Maryland, USA, pp. 152–158. Association for Computational Linguistics.
- Mihalcea, R. F. and D. I. Moldovan (2001). A highly accurate bootstrapping algorithm for word sense disambiguation. *International Journal on Artificial Intelligence Tools Vol. 10*, pp. 5–21.
- Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM Vol. 38*, pp. 39–41.
- Mitkov, R. (2004). *The Oxford Handbook of Computational Linguistics*. Oxford, UK: Oxford University Press.
- Nguyen, K.-H. and C.-Y. Ock (2011). Word sense disambiguation as a traveling salesman problem. *Artificial Intelligence Review*, pp. 1–23.
- Noy, N. F. and D. L. McGuinness (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Technical report, Stanford University.
- Oliveira, H. G. and P. Gomes (2012). Integrating lexical-semantic knowledge to build a public lexical ontology for Portuguese. In *Natural Language Processing and Information Systems, Proceedings of 17th NLDB*, Vol. 7337 of *LNCS*, Groningen, The Netherlands, pp. 210–215. Springer.

- Patwardhan, S., S. Banerjee, and T. Pedersen (2003). Using measures of semantic relatedness for word sense disambiguation. In *Proceedings of 4th International Conference on Intelligent Text Processing and Computational Linguistics*.
- Patwardhan, S., S. Banerjee, and T. Pedersen (2005). Sensesrelate::targetword – a generalized framework for word sense disambiguation. In *Proceedings of Association for Computational Linguistics*, pp. 73–76.
- Piruzelli, M. P. F. and B. C. Dias da Silva (2010). Estudo exploratório de informações lexicais relevantes para a resolução de ambiguidades lexical e estrutural. In *Anais do Encontro do Círculo de Estudos Linguísticos do Sul*, Universidade do Sul de Santa Catarina, Palhoça, SC.
- Plaza, L. and A. Diaz (2011). Using semantic graphs and word sense disambiguation techniques to improve text summarization. In *Proceedings of Procesamiento del Lenguaje Natural*, Vol. 47, pp. 97–105.
- Radev, D. R. (2000). A common theory of information fusion from multiple text sources step one: Cross-document structure. In *Proceedings of 1st ACL SIGDIAL Workshop on Discourse and Dialogue*, Hong Kong–China.
- Ratnaparkhi, A. (1986). A maximum entropy model for part-of-speech tagging.
- Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 1*, IJCAI’95, San Francisco, CA, USA, pp. 448–453. Morgan Kaufmann Publishers Inc.
- Santos, D. (2008). *Avanços da Linguística de Corpus do Brasil*, Chapter Corporizando Algumas Questões, pp. 41–66. Humanitas.
- Sardinha, T. B. (2005). *A Língua Portuguesa no Computador*. Mercado de Letras.
- Silva, T. C. and D. R. Amancio (2012). Word sense disambiguation via high order of learning in complex networks. *EPL (Europhysics Letters)* Vol. 98 N. 5, pp. 58001.
- Sinha, R. and R. Mihalcea (2007). Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of IEEE International Conference on Semantic Computing*.
- Specia, L. (2007). *Uma Abordagem Híbrida Relacional para a Desambiguação Lexical de Sentido na Tradução Automática*. Ph. D. thesis, Instituto de Ciências Matemáticas e de Computação–ICMC–USP.

- Turney, P. (2004). Word sense disambiguation by web mining for word co-occurrence probabilities. In R. Mihalcea and P. Edmonds (Eds.), *Proceedings of Senseval-3: 3th International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, pp. 239–242. Association for Computational Linguistics.
- Vasilescu, F., P. Langlais, and G. Lapalme (2004). Evaluating variants of the lesk approach for disambiguating words. In *Proceedings of Language Resources and Evaluation (LREC 2004)*, Lisbonne, Portugal, pp. 633–636.
- Veronis, J. (1998). A study of polysemy judgements and inter-annotator agreement. In *Proceedings of Programme and Advanced Papers of the Senseval Workshop, Herstmonceux*, pp. 2–4.
- Vossen, P. (Ed.) (1998). *EuroWordNet: a multilingual database with lexical semantic networks for European Languages*. kluwer.
- Weischedel, R., S. Pradhan, L. Ramshaw, J. Kaufman, M. Franchini, and M. El-Bachouti (2010). Ontonotes release 4.0. online. Available in: <http://www ldc.upenn.edu/Catalog/docs/LDC2011T03/OntoNotes-Release-4.0.pdf>.
- Wilks, Y. and M. Stevenson (1997). Combining independent knowledge sources for word sense disambiguation. In *Proceedings of RANLP*, pp. 1–7.
- Witten, I. H. and E. Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2 ed.). Morgan Kaufmann series in data management systems. Morgan Kaufmann.
- Wu, Z. and M. Palmer (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics, ACL '94*, Stroudsburg, PA, USA, pp. 133–138. Association for Computational Linguistics.
- Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196.
- Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley (Reading MA).

Apêndice: Textos de Treinamento

Neste Apêndice, é apresentada a coleção de documentos utilizada para o treinamento dos anotadores durante o processo de anotação do corpus CSTNews, descrito no Capítulo 4. Esta coleção de documentos é composta por dois textos jornalísticos, que abordam um mesmo assunto (BC corta juro em 0,5 ponto porcentual, para 8,5% ao ano) e foram extraídos de jornais eletrônicos distintos, o Estadão¹ e o Jornal Folha de São Paulo², cujos textos são apresentados, respectivamente, nas Seções A.1 e A.2.

A.1 Estadão

BRASÍLIA – O Banco Central cortou nesta quarta-feira, 30, o juro básico da economia pela sétima vez consecutiva. O tamanho do corte, porém, diminuiu para 0,5 ponto. Com isso, a Selic caiu para 8,5%, o menor nível da história. Em março e abril, os cortes haviam sido de 0,75 ponto. Antes, o menor juro foi de 8,75% entre julho de 2009 e abril de 2010.

A decisão foi unânime. Além de baratear empréstimos, a taxa menor também dispara o gatilho que muda o cálculo da remuneração das novas cadernetas de poupança. A partir de agora, elas passam a pagar rendimento de 70% da Selic. A regra vale para depósitos a partir de 4 de maio.

Ao explicar a decisão, os diretores do BC afirmam que “neste momento, permanecem limitados os riscos para a trajetória da inflação”. Além disso, “dada a fragilidade da economia global, a contribuição do setor externo tem sido desinflacionária”.

¹<http://www.estadao.com.br/>

²<http://www.folha.uol.com.br/>

No comunicado distribuído após o anúncio da decisão do Comitê de Política Monetária (Copom), o BC explicitou o nome dos seis diretores e do presidente Alexandre Tombini como os que apoiaram a redução em 0,5 ponto - nova regra de transparência adotada pelo governo. (Leia abaixo)

O fraco desempenho da economia e a preocupação com o nebuloso cenário externo estão por trás da decisão do BC de continuar com os cortes. Mas o patamar inédito da taxa exige atenção redobrada dos diretores com o efeito acumulado dos cortes desde agosto e, por isso, o Copom decidiu diminuir o ritmo.

Estoques elevados, lenta reação da indústria e início de demissões em algumas empresas são algumas das notícias no Brasil que mostram que a economia deve demorar mais tempo que o imaginado para acelerar. Nesta semana, o mercado passou a apostar pela primeira vez que o País deve crescer menos de 3% neste ano. Enquanto isso, as previsões para a inflação não sinalizam preocupação.

No exterior, a situação é ainda mais preocupante. Com o agravamento da crise na Europa, há possibilidade de que a Grécia deixe a área do Euro. Houve, ainda, divulgação de problemas em algumas instituições financeiras europeias. Além disso, há sinais de que a China deve crescer menos, o que reduziu os preços das commodities, segmento em que o Brasil é grande exportador.

A.2 Folha de São Paulo

O Copom (Comitê de Política Monetária do Banco Central) anunciou nesta quarta-feira (30) a redução de 0,5 ponto percentual na taxa básica de juros, a Selic.

Com a decisão, a taxa caiu de 9% para 8,5% ao ano – o menor patamar da série histórica iniciada em 1986 –, reduzindo o rendimento da poupança, que cai a partir de amanhã.

A nova regra determina que com a taxa abaixo de 8,5%, um gatilho será acionado e as novas cadernetas de poupança e novos depósitos terão seus rendimentos calculados com base em 70% da Selic, acrescidos da TR (Taxa Referencial, que não muda). A regra vale apenas para depósitos feitos a partir de 4 de maio.

Além de atingir o piso histórico da taxa, a divulgação de hoje é a primeira em que foi detalhada como votou cada um dos sete membros do comitê, decisão baseada na Lei de Acesso à Informação. A decisão, segundo o Copom, foi unânime.

Votaram pela redução da taxa Selic para 8,50% os seguintes membros do comitê: Alexandre Antonio Tombini, presidente do Banco Central, Aldo Luiz Mendes, Altamir Lopes, Anthero de Moraes Meirelles, Carlos Hamilton Vasconcelos Araújo, Luiz Awazu Pereira da Silva e Sidnei Corrêa Marques.

A decisão de baixar os juros em 0,5 p.p. era esperada pelo mercado. Desde o início do ano, o ambiente econômico, com fraco crescimento da economia, indicavam que o Copom manteria sua estratégia de redução de juros. Esta é a sétima vez consecutiva que a autoridade monetária reduz os juros.

Na sexta-feira (1), o IBGE irá divulgar o resultado do crescimento do PIB (Produto Interno Bruto) no primeiro trimestre. As estimativas são de que o número não deve ficar acima de 0,5% em relação ao trimestre anterior. Em 2011, o crescimento do PIB foi de 2,7%.

Apesar da estimativa de uma ligeira recuperação do crescimento em abril e maio, a equipe econômica já trabalha com um número ruim no segundo trimestre, levemente superior ao do primeiro, o que deve levar a um crescimento de cerca de 3% no ano.

Apêndice: Resumo de Anotação

Neste Apêndice, é apresentada uma tabela com o resumo da anotação do corpus CST-News. Nesta, para cada palavra anotada, são apresentadas as informações referentes a: frequência de ocorrência de instâncias anotadas; número de traduções empregadas; número de possíveis *synsets*; e número de *synset* diferentes anotados no corpus.

Por exemplo, para a palavra “ação”, tem-se que foram anotadas 6 ocorrência (valor da frequência), sendo empregadas apenas um tradução e dois *synsets*, sendo 10 possibilidades de desambiguação (quantidade de possíveis *synsets*).

Palavra	Frequência	N Traduções	<i>Synsets</i> Possíveis	<i>Synsets</i> Anotados
abertura	6	1	13	1
abrigo	5	3	9	1
ação	6	1	10	2
acaso	2	2	7	1
acidente	7	1	2	2
aeronave	3	2	2	1
aeroporto	32	1	1	1
agência	13	1	5	3
água	19	1	6	2
ajuda	6	4	28	1
alagamento	5	2	4	1
alimento	7	2	4	1
aluno	5	1	3	1
alvo	4	2	8	2

andamento	3	2	11	1
andar	5	2	9	1
ano	92	1	4	3
apreensão	10	2	5	1
apresentador	12	1	3	1
área	18	7	54	4
argentino	6	0	0	1
arquivamento	5	1	4	1
artigo	6	2	13	1
artista	3	1	1	1
assalto	5	3	15	1
assessor	2	3	4	1
astrônomo	4	1	1	1
ataque	36	3	25	2
atirador	4	2	5	1
atleta	6	1	1	1
atraso	10	3	9	1
aula	8	2	12	1
aumento	9	5	21	2
autuação	11	2	4	1
avenida	10	1	2	1
avião	13	2	5	1
banco	12	1	10	1
base	4	2	30	1
biocombustíveis	5	0	0	1
bola	22	1	12	1
bomba	12	2	8	1
bombardeio	3	1	4	1
busca	9	3	10	1
cabeceira	6	3	6	1
campus	16	0	0	1
cancelamento	6	1	2	1
candidato	25	1	2	1
canteiro	5	2	0	1
capital	7	1	8	1
cara	4	4	29	3
cargo	6	8	59	1
carteira	11	4	7	1

cartel	5	0	0	1
casa	21	1	12	1
caso	6	4	24	1
categoria	5	1	2	1
centro	17	4	29	3
chegada	8	1	3	1
chute	5	2	8	1
chuva	26	1	3	1
cidade	34	2	7	1
cobrança	4	2	19	1
cocaína	4	1	1	1
coisa	4	1	12	1
colocação	4	3	11	1
colombiano	5	0	0	1
companhia	9	2	12	1
comparação	8	1	3	2
competição	16	1	4	3
complicação	4	2	11	1
conferência	3	4	15	1
conflito	7	4	22	2
congestionamento	4	1	2	1
conquista	9	4	5	1
consciência	4	4	11	1
conselheiro	4	3	5	1
conselho	10	4	16	2
contribuinte	18	1	2	1
controle	10	1	11	1
corpo	4	1	11	1
costa	5	3	15	1
craque	4	1	1	1
crédito	12	1	9	2
crescimento	10	2	11	2
criança	3	3	7	1
critério	6	1	2	1
cronograma	4	1	0	1
dado	4	5	3	1
dano	5	1	5	2
decisão	7	1	5	2

declaração	15	1	6	2
decolagem	8	0	0	1
decoro	3	2	3	1
defeito	1	5	15	1
degradação	3	1	2	1
denúncia	15	1	1	2
deputado	36	2	8	1
desabrigados	4	2	0	1
desaparecido	4	1	0	1
desembargador	7	0	0	1
desenvolvimento	8	3	18	1
desmatamento	2	1	2	1
dia	33	1	10	2
dinheiro	9	1	3	1
direito	6	2	8	1
diretor	5	3	43	1
diretoria	4	4	17	1
discurso	10	1	8	2
documento	14	1	4	1
domingo	5	0	0	1
dormitório	6	1	2	1
drible	6	1	3	1
droga	5	1	1	1
economista	4	1	1	1
efeito	5	1	6	1
empresa	16	2	10	1
empréstimo	3	2	4	1
enchente	7	1	6	1
energia	3	3	25	1
entrada	4	1	3	1
envio	7	1	2	1
envolvimento	10	1	5	2
epicentro	4	1	1	1
equipe	16	1	2	1
escândalo	4	1	2	1
escola	9	1	7	2
estabilidade	3	1	3	1
estado	12	1	8	3

estréia	4	3	14	1
estrela	8	1	8	1
estudante	9	1	2	1
estudo	4	3	12	1
etanol	2	1	1	1
etapa	8	1	8	1
evento	5	3	6	1
exemplo	2	3	13	1
exército	7	3	9	1
fábrica	4	2	5	1
falho	7	3	1	2
falto	1	3	1	1
família	4	1	8	1
fase	4	1	4	1
favela	3	2	1	1
favorito	4	3	7	1
ferido	4	3	21	1
ferrovia	3	2	2	1
festa	5	1	5	1
fila	4	3	40	1
filho	4	1	2	3
final	8	1	2	1
financiamento	3	1	1	1
fiscalização	10	3	13	1
foguete	5	2	6	1
fome	2	1	2	1
fonte	5	2	13	2
força	4	5	34	1
funcionário	5	3	5	1
fundo	5	3	19	2
furacão	24	1	1	1
futebol	6	2	3	1
gado	5	4	21	1
ganho	3	4	7	1
garçom	4	2	5	1
gasoduto	3	1	0	1
gesto	1	2	4	1
gol	26	1	4	2

goleiro	7	1	2	1
governo	26	1	4	1
grafiteiro	4	0	0	1
grau	4	2	15	1
grupo	4	1	3	1
guerra	7	1	4	1
guerrilha	5	1	1	1
h	9	1	0	1
habitação	3	2	3	1
hepatite	4	1	1	1
história	4	2	10	1
hora	4	1	4	3
horário	9	2	4	2
humildade	4	2	3	1
ilha	10	2	3	1
imposto	4	1	1	1
inadimplência	3	0	0	1
indício	4	4	15	1
infarto	4	0	0	1
informação	11	1	5	1
infra-estrutura	8	1	2	1
instituição	3	1	5	1
insuficiência	6	2	3	1
intenção	14	1	3	1
inundação	13	3	7	1
investigação	32	1	2	3
investimento	8	0	0	2
irmão	4	1	5	1
janeiro	6	0	0	1
jogada	8	2	9	1
jogador	10	2	7	1
jogo	14	5	41	2
jóia	5	2	5	1
jornal	6	1	4	1
juiz	4	0	0	1
juízo	5	2	0	1
julho	18	0	0	1
junho	13	0	0	1

km	13	1	1	1
lentidão	6	1	3	1
levantador	6	0	0	1
licenciamento	3	0	0	1
lista	5	2	4	1
local	7	4	23	2
lucro	20	2	5	1
m	43	2	0	2
madrugada	5	3	1	1
máfia	4	0	0	1
magnitude	8	1	3	1
malha	15	2	5	1
mandado	4	3	19	1
mandato	9	3	24	1
manhã	15	1	4	1
manobra	4	2	5	1
marca	15	2	20	2
marcação	4	0	0	1
março	11	0	0	1
massa	8	3	14	1
material	6	1	5	1
medalha	23	1	1	1
medida	4	2	9	1
médio	6	3	14	1
melhoria	3	2	5	1
membro	4	1	5	1
mensalão	4	0	0	1
mês	13	1	2	1
meta	1	4	16	1
metade	1	1	2	1
metro	13	2	4	1
micro	3	0	0	1
miliciano	4	2	1	1
min	13	0	0	1
ministério	1	1	4	1
ministro	4	1	4	1
minuto	15	0	0	1
missão	2	1	5	1

míssil	7	1	2	1
momento	4	2	7	1
morte	5	1	8	2
morto	9	3	3	1
motim	4	3	3	1
moto	7	2	2	1
mudança	1	3	22	1
mundo	8	1	8	2
mural	7	2	1	1
nadador	5	2	3	1
natação	5	1	1	1
nível	6	3	13	1
noite	4	1	8	1
nome	4	4	21	1
notícia	1	3	13	1
número	13	1	11	2
obra	40	2	8	4
ocorrência	5	2	2	1
omissão	6	2	9	1
ônibus	9	1	4	1
operação	9	1	11	2
oposição	7	1	8	2
organização	8	2	10	1
ouro	21	1	5	3
pagamento	4	2	5	1
país	43	2	7	3
par	5	3	10	1
paralisação	4	1	0	1
parciais	4	0	0	1
parlamentar	20	3	2	2
parte	10	1	12	5
participação	6	1	2	1
partida	26	3	23	3
partido	9	2	5	1
passageiro	4	2	2	1
passagem	2	4	29	1
paz	6	1	5	1
pedido	17	3	6	2

peemedebista	3	0	0	1
pena	4	2	5	1
península	3	1	1	1
perícia	7	1	1	1
período	21	1	7	2
pesquisa	19	2	5	1
pessoa	67	1	3	2
pista	32	1	11	2
planeta	6	1	3	1
pobreza	3	1	1	1
polícia	35	1	1	1
policial	5	0	0	1
política	6	2	8	1
ponto	11	5	37	2
população	3	1	5	1
porta-voz	7	1	1	1
posto	6	3	38	1
pouso	12	0	0	2
povo	6	2	8	1
prazo	5	2	8	1
prédio	8	1	4	1
prefeito	4	1	1	1
presidente	79	1	6	5
presídio	5	2	2	1
preso	15	3	5	1
prioridade	4	1	2	1
prisão	4	0	0	1
problema	5	3	14	2
processo	34	1	6	1
procurador	12	2	3	2
produção	4	1	8	1
programa	3	1	8	1
proposto	7	1	0	1
prorrogação	5	2	2	1
prova	4	3	15	1
quadra	4	2	16	1
quadrilha	5	0	0	1
quarto	8	4	21	2

quebra	3	3	7	1
quinto	5	4	15	1
reação	5	1	7	1
rebelde	4	5	6	1
rebelião	7	3	3	1
recaída	4	3	2	1
recorde	12	3	11	1
recurso	7	2	7	1
reforma	16	3	8	1
região	14	3	10	1
regime	5	4	12	1
registro	4	3	20	1
relator	22	3	2	2
relógio	6	2	7	1
renda	6	3	5	1
renúncia	14	2	7	1
reportagem	1	2	0	1
representação	28	1	10	2
requerimento	5	2	7	1
resolução	4	1	11	1
responsabilidade	2	1	3	1
resultado	16	3	11	3
reunião	13	2	8	1
reverso	1	3	9	1
reversor	2	0	0	1
revezamento	13	3	9	2
revista	5	3	13	1
rio	6	1	1	1
risco	3	4	19	1
roubo	11	2	3	1
rua	8	1	5	1
s	22	0	0	1
saída	4	1	3	1
sala	4	3	22	2
salto	12	6	14	2
saneamento	8	1	2	1
sanguessuga	9	2	2	1
saúde	6	2	4	1

segunda-feira	23	0	0	1
segundo	4	2	10	2
segurança	5	2	15	2
seleção	38	3	5	1
semáforo	4	3	2	1
semana	14	1	3	1
semestre	7	1	2	1
senador	52	0	0	1
sentido	15	2	7	1
ser	5	2	3	1
sessão	6	1	4	1
set	21	0	0	1
sexta-feira	13	0	0	1
sexto	4	2	3	2
sistema	10	1	9	2
sobrestamento	6	0	0	1
sul	4	1	5	1
suspeito	6	3	2	1
técnico	7	2	7	1
tempestade	3	2	4	1
tempo	14	2	11	3
tentativo	5	2	6	1
terça	4	1	0	1
terça-feira	15	0	0	1
terra	1	2	17	1
terremoto	29	1	2	1
time	14	1	2	1
tiro	9	0	0	1
tiroteio	9	3	7	1
título	17	4	23	1
tocha	4	3	10	1
torcedor	4	4	5	1
torcida	11	1	1	1
trabalho	6	1	7	1
tráfego	11	1	4	1
traficante	8	4	7	1
tráfico	5	3	11	1
tragédia	4	1	2	1

trânsito	4	2	7	1
transporte	5	2	10	1
trecho	4	5	38	1
treinador	4	2	2	1
tremor	17	3	6	1
trimestre	9	2	16	1
tripulação	4	2	6	1
tropa	4	0	0	1
tucano	9	0	0	1
turista	4	1	1	1
turno	20	2	22	2
unidade	6	1	6	1
unificação	7	2	6	1
universidade	17	1	3	1
urbanização	3	1	2	1
usina	16	5	21	1
vaia	18	3	6	1
valor	6	1	6	1
vazamento	8	2	5	1
venda	9	3	6	1
vento	9	1	8	1
vez	19	2	22	3
viagem	3	2	9	1
vitória	12	1	1	1
vôlei	8	1	2	1
vôo	28	1	9	3
votação	12	3	5	1
voto	17	0	0	1
zona	2	1	4	1