Proceedings of Symposium in Information and Human Language Technology. Natal, RN, Brazil, November 4–7, 2015.

Desambiguação de Homógrafos–Heterófonos por Aprendizado de Máquina em Português Brasileiro

Leonardo Hamada[1], **Nelson Neto**[1]

[1]Instituto de Ciências Exatas e Naturais
Universidade Federal do Pará (UFPA) – Belém, PA – Brasil

hamadaleonardo@gmail.com, nelsonneto@ufpa.br

Abstract. *To improve the quality of the speech produced by a text-to-speech system, it is important to obtain the maximum amount of information from the input text that may help in this task. In this context, the word sense disambiguation plays an important role and still be a central problem for natural language processing applications. This paper proposes to model the ambiguity of words as a supervised machine learning problem for Brazilian Portuguese. In doing so, four algorithms (or classifiers) were compared in two types of texts. Computer experiments showed that to assure portability of systems, a process of tuning to the new domain is required.*

Resumo. *Para aprimorar a qualidade da voz produzida por um sistema de conversão texto-fala, é importante extrair a maior quantidade possível de informação, que possa ajudar nessa tarefa, a partir do texto de entrada. Nesse contexto, a desambiguação da pronúncia relativa a pares de homógrafos-heterófonos (HHs) assume um papel relevante e ainda de difícil tratamento em aplicações que envolvem processamento de linguagem natural. Este trabalho propõe modelar a ambiguidade entre HHs falados no Brasil como um problema de aprendizado de máquina supervisionado. Para isso, quatro algoritmos (ou classificadores) foram comparados em bases de texto de diferentes tipos. Experimentos mostraram que para garantir a portabilidade de sistemas, um processo de incremento para o novo domínio é necessário.*

1. Introdução

Segundo [Cardie 1996], o problema de ambiguidade entre palavras pode ser genericamente caracterizado da seguinte forma: "Em um dado momento, os sistemas de processamento de linguagem natural recebem um segmento de informação que pode ter várias interpretações, e ele precisa decidir qual interpretação é a mais apropriada para aquele contexto. A fim de resolver essa dificuldade, é necessário desambiguar semanticamente, sintaticamente ou estruturalmente duas ou mais formas distintas com base nas propriedades que circundam o contexto".

Por exemplo, na frase: "Tenho uma **cerca** na minha casa. Ela **cerca** toda a área e tem **cerca** de oito metros", percebe-se três situações possíveis de ocorrência do homógrafo-heterófono (HH) "cerca": substantivo ("c[e]rca"), verbo ("c[E]rca") e preposição ("c[e]rca"), respectivamente. Logo, o desambiguador de HH deve ser capaz de determinar a correta transcrição fonética em cada situação.

Mesmo que o número de HHs existente represente um percentual bem pequeno em relação ao texto analisado, no contexto de síntese de voz (TTS ou *text-to-speech*), a transcrição fonética equivocada tem consequência direta na qualidade da voz gerada, atraindo a atenção do ouvinte para o erro corrente. Diminuir os erros de pronúncia entre HHs melhora significativamente a naturalidade e a inteligibilidade do sintetizador de voz [Ribeiro et al. 2009]. Diante do exposto, é fundamental que a desambiguação de HHs faça parte do conjunto de algoritmos responsável pela transcrição fonética dentro de um sistema TTS, ou seja, é importante a presença de um recurso que decida qual será a tonicidade (i.e. vogal tônica aberta ou fechada) da vogal diferencial do HH.

Assim, o objetivo desta pesquisa é empregar técnicas de aprendizado de máquina para tratar a desambiguação de HHs em Português Brasileiro (PB). Diferentemente da maioria dos trabalhos nessa linha, a ideia não é elaborar um conjunto de regras linguísticas, mas sim explorar o uso de classificadores inteligentes construídos a partir de treinamento supervisionado para resolver o problema de ambiguidade entre palavras. Este estudo terá aplicações práticas em um sistema real de conversão texto-fala, além de avaliar a portabilidade e afinação de diferentes algoritmos de aprendizado de máquina através do seu treinamento e teste em bases de dados de diferentes domínios.

2. Revisão Bibliográfica

Sobre desambiguação de HHs em sistemas TTS para o PB, [Seara et al. 2001, Seara et al. 2002] mostram um analisador morfossintático para solucionar o problema de alternâncias vocálicas entre substantivos e verbos, sem, no entanto, abordar a desambiguação de HHs semanticamente. Outros trabalhos têm ambas as abordagens, morfossintática e semântica, porém, as gramáticas implementadas foram testadas apenas com um ou dois exemplos de HHs [Ferrari et al. 2003, Barbosa et al. 2003a, Barbosa et al. 2003b].

Em [Shulby et al. 2013], os autores apresentam duas regras, específicas as pronúncias das vogais <e> e <o>, para desambiguar 226 pares de HHs. Para a vogal <e>, a transcrição fonética será [E] (aberta) quando o HH for classificado como verbo, e [e] (fechada), quando o HH for classificado como substantivo e a vogal <e> estiver na sílaba tônica. Para a vogal <o>, a regra de transcrição fonética é similar. O trabalho apresenta até 95% de acerto em alguns pares, porém, limita-se apenas a duas categorias de HHs (verbo e substantivo).

[Silva et al. 2012] propôs que a análise morfossintática seria suficiente para desambiguar HHs pertencentes a classes gramaticais distintas; e, para os HHs de mesma classe gramatical, uma análise semântica seria necessária. O trabalho resultou em 23 algoritmos de desambiguação para um conjunto de 111 pares de HHs. Apesar dos algoritmos estarem publicados, [Silva et al. 2012] não incluiu detalhes precisos sobre as bibliotecas gramaticais utilizadas, o que dificulta a implementação desses algoritmos.

Uma das linhas de pesquisa atuais de maior sucesso é a abordagem baseada em *data-driven*, nas quais algoritmos estatísticos ou de aprendizado de máquina têm sido aplicados para construir modelos estatísticos ou classificadores a partir de informações extraídas de grandes bases de texto (comumente chamadas na literatura de corpus), no intuito de resolver o problema da desambiguação de palavras.

Já há algum tempo, o método *data-driven* vem sendo bastante explorado pela comunidade científica dada a sua importância na área de síntese de voz em diversos idiomas.

Em [Yarowsky 1997], os autores apresentam uma tipologia de HHs na língua inglesa e algumas técnicas tradicionalmente usadas na desambiguação, tais como N-gram *taggers*, classificadores bayesianos e árvores de decisão, bem como a proposta de um sistema híbrido, ao combinar as técnicas descritas. Tal interesse também é visto em línguas de menor expressão, como o Português Europeu (PE), por exemplo. Em [Ribeiro et al. 2003], um desambiguador que mescla regras linguísticas e modelos probabilísticos de Markov é descrito, e a influência das informações morfossintáticas na tarefa de desambiguação é analisada dentro de um sistema TTS para o PE.

Normalmente, o método de aprendizado utilizado é o supervisionado, onde o classificador ou modelo estatístico é treinado a partir de bases de texto previamente anotadas (ou etiquetadas) sintaticamente e/ou morfologicamente. Além da sabida carência de extensas bases de texto etiquetadas de forma confiável, especialmente para o PB, a abordagem supervisionada apresenta outra particularidade: a desambiguação de HHs é extremamente dependente do domínio da aplicação, como afirma [Màrquez 2000]. Em outras palavras, não parece razoável pensar que o material de treinamento é grande e representativo o suficiente para cobrir todos os tipos possíveis de amostras. Adicionalmente, é preciso estudar até que ponto um treinamento tendo como base um texto jornalístico pode ser portado para um texto literário, por exemplo. Até onde se pesquisou, essa estratégia de desambiguação ainda não foi abordada para o PB.

3. Materiais e Métodos

A classificação é uma tarefa onde um modelo é construído para prever uma categoria, como, por exemplo, "sim" ou "não", "aberta" ou "fechada". Para que os algoritmos de classificação funcionem, é necessário separar o processo em duas partes: treino, onde o algoritmo analisa os dados de treinamento; e a classificação propriamente dita, onde dados de teste são usados para estimar a acurácia dos algoritmos [Han et al. 2011, p. 328]. Dessa forma, a desambiguação de palavras pode ser facilmente formulada como um problema de classificação supervisionada, ou seja, conhecimento extraído a partir de textos.

Os classificadores usados neste trabalho para desambiguação de HHs foram selecionados explorando a biblioteca de implementações do ambiente WEKA versão 3.6.11 [Hall et al. 2009], mantendo os parâmetros padrões dos algoritmos. Isto posto, os algoritmos escolhidos e os motivos foram:

- **(i)** Naive Bayes: simples e de natureza estatística, é um algoritmo clássico para resolver ambiguidade em outras línguas. Utiliza o teorema de Bayes e pressupõe independência de atributos [Bielza and Larrañaga 2014];
- **(ii)** AODE: também de natureza estatística, busca melhorar o Naive Bayes ao relaxar as suposições de independência [Bielza and Larrañaga 2014];
- **(iii)** J48: é uma árvore de decisão que implementa o algoritmo C4.5. É possível visualizar a árvore gerada pela indução de regras [Witten and Frank 2005, p. 198];
- **(iv)** Random Forest: utiliza várias árvores de decisão elegendo a resposta por voto majoritário e ameniza o problema de *overfitting* durante o treino [Witten and Frank 2005, p. 407].

Normalmente, cada HH é tratado como um problema de classificação diferente. Portanto, a coletânea de um corpus representativo deve considerar as particularidades de cada palavra, a fim de decidir o número de exemplos necessários para aprendizagem, os

atributos mais úteis, e assim por diante. Outra dificuldade é a aquisição de uma base de conhecimento corretamente etiquetada morfologicamente e/ou sintaticamente.

Para este trabalho, formulou-se dois corpora: um com textos jornalísticos (corpus A) e outro com textos literários (corpus B). A partir dessas bases de dados, localizou-se as frases que continham HHs existentes no português falado no Brasil para formar os conjuntos de treino e avaliação. Observou-se, no entanto, que os pares de pronúncias ocorrem muitas vezes de forma desbalanceada. Assim, optou-se pelos HHs que apresentaram as menores diferenças entre suas ocorrências aberta e fechada, conforme a Tabela 1.

Tabela 1. Distribuição dos HHs selecionados a partir dos corpus A e B.

Palavra	Corpus A			Corpus B		
	Abertas	Fechadas	Ocorrências	Abertas	Fechadas	Ocorrências
"colher"	81	247	328	11	83	94
"corte"	109	104	213	3	24	27
"fora"	573	28	601	198	67	265
"gosto"	140	108	248	58	337	395
"começo"	15	391	406	18	68	86
"rola"	116	9	125	17	18	35
"sede"	118	7	125	12	38	50
Total	1152	894	2046	317	635	952

Em seguida, o anotador morfossintático MXPOST foi usado para etiquetar as frases escolhidas. O MXPOST é baseado na técnica de máxima entropia e foi inicialmente disponibilizado para a língua inglesa, sendo adaptado para o PB por [Aires et al. 2000], tendo o corpus Mac-Morpho como sua base de treino. Por fim, o vetor de atributos de cada frase (conteúdo do arquivo de entrada do WEKA) foi gerado a partir de um *script* automático, sendo a vogal diferencial do HH classificada manualmente como aberta ou fechada. As bases de texto e o vetor de atributos serão melhor detalhados a seguir.

3.1. Corpus A

O corpus A é composto pelos seguintes conjuntos de textos predominantemente de natureza jornalística:

- **(i)** Corpus Mac-Morpho revisado, composto de textos jornalísticos extraídos de dez seções do jornal diário Folha de São Paulo do ano de 1994, contendo cerca de um milhão de palavras [Aluísio et al. 2003, Fonseca and Rosa 2013];
- **(ii)** Corpus CETEN-Folha, composto de textos com cerca de 24 milhões de palavras extraídos do jornal Folha de S. Paulo e compilado pelo [NILC 2002] da USP;
- **(iii)** Texto da Constituição da República Federativa do Brasil de 1988 [Brasil 1988];
- **(iv)** Aproximadamente 25% dos artigos em português da enciclopédia Wikipédia[1];
- **(v)** Corpus LapsNEWS, uma coletânea de textos jornalísticos retirados de dez jornais brasileiros disponíveis na Internet, contendo aproximadamente 120 mil frases [Neto et al. 2011].

[1]Conteúdo obtido em 20 de fevereiro de 2015 na página http://dumps.wikimedia.org/ptwiki/

3.2. Corpus B

O corpus B é composto pelas seguintes obras literárias (escritor e quantidade): José de Alencar (21); Machado de Assis (11); Olavo Bilac (149); Castro Alves (62); e Euclides da Cunha (5) [ABL 2011, Portal S. F. 1998]. Também utilizou-se o corpus eletrônico anotado Tycho Brahe [Galves and Faria 2010], composto de 66 textos escritos por autores nascidos entre 1380 e 1881.

3.3. Vetor de Atributos

Para gerar o arquivo ARFF conformante para processamento pelo WEKA, foi necessário definir quais tipos de atributos deveriam ser armazenados para formar a base de treino dos classificadores. Adotou-se um modelo para o vetor de atributos, um por frase, apresentado em [Màrquez 2000] para contextos locais:

$$p_{-3},\ p_{-2},\ p_{-1},\ p_{+1},\ p_{+2},\ p_{+3},\ w_{-1},\ w_{+1},\ (w_{-2},\ w_{-1}),\ (w_{-1},\ w_{+1}),\ (w_{+1},\ w_{+2}),$$
$$(w_{-3},\ w_{-2},\ w_{-1}),\ (w_{-2},\ w_{-1},\ w_{+1}),\ (w_{-1},\ w_{+1},\ w_{+2}),\ (w_{+1},\ w_{+2},\ w_{+3})$$

onde $w_{\pm 3}$ é o contexto de palavras consecutivas ao redor da palavra w a ser desambiguada, e $p_{\pm 3}$ é a etiqueta fornecida pelo MXPOST para a palavra $w_{\pm 3}$. Ao todo são 15 atributos. Os arquivos ARFFs foram gerados automaticamente através de um *script* escrito na linguagem Python. Abaixo, dois exemplos (frase e vetor de atributos) são apresentados.

i) O governador eleito almoçou uma **colher** de arroz e 40 gramas de carne

```
ADJ,VERB,ART,PREP,N,CONJ,uma,de,almoçou_uma,uma_de,
de_arroz,eleito_almoçou_uma,almoçou_uma_de,
uma_de_arroz,de_arroz_e,1
```

ii) Os fiscais vão **colher** amostras para análise

```
ART,N,VERB,N,PREP,N,vão,amostras,fiscais_vão,vão_amostras,
amostras_para,Os_fiscais_vão,fiscais_vão_amostras,
vão_amostras_para,amostras_para_análise,0
```

Além dos atributos, o vetor contém um campo binário, chamado classe, que marca a tonicidade da vogal diferencial do HH presente na frase ("1" para aberta e "0" para fechada). Essa marcação foi realizada manualmente em uma interface *Web*[2] implementada especificamente para esta tarefa, usando a linguagem Lua e o banco de dados Sqlite3[3].

4. Experimentos

A comparação entre os algoritmos foi realizada através de uma série de experimentos controlados usando exatamente os mesmos conjuntos de treino e teste. Também visando avaliar a dependência da desambiguação de HHs com relação ao domínio da aplicação, foram elaboradas sete combinações possíveis para os conjuntos treino-teste. Por exemplo, a notação A–B indica que o algoritmo foi treinado com o corpus A e avaliado com o corpus B, assim como a notação A+B–B diz que formou-se a base de treino com a união dos corpora A e B, e a base de teste apenas com corpus B.

[2] Acessível em http://homografos.ddns.net:81/fcgi-bin-r/index.lua

[3] Lua, Python e Sqlite acessíveis em http://www.lua.org, https://www.python.org e http://www.sqlite.org

4.1. Primeiro Experimento

A Tabela 2 apresenta a média de acertos dos quatro algoritmos para todas as combinações dos conjuntos treino-teste. Utilizou-se o teste cruzado em 10-*folds*, exceto nos casos A–B e B–A. O ZeroR é um algoritmo de referência do WEKA em que a pronúncia mais frequente no conjunto de treino é usada para classificar todos os exemplos presentes no conjunto de teste. O melhor resultado para cada caso é destacado em negrito.

Tabela 2. Acurácia dos algoritmos para todas as combinações de treino-teste.

Algoritmo	Acurácia (%)						
	A+B—A+B	A+B—A	A+B—B	A—A	B—B	A—B	B—A
ZeroR	79,92	80,35	79,92	80,65	80,56	46,11	62,64
Naive Bayes	90,56	90,86	90,43	90,04	89,38	**82,35**	**86,12**
AODE	**94,16**	**94,55**	**93,93**	**94,34**	**94,10**	78,99	83,32
J48	91,49	92,49	91,49	92,29	91,66	77,42	74,35
RandomForest	88,76	90,34	89,04	89,83	89,20	58,72	69,75

Observou-se que o AODE superou os outros algoritmos, exceto nas combinações A–B e B–A, onde prevaleceu o algoritmo Naive Bayes simples. Nesses casos em específico, os conjuntos de treino e teste são totalmente disjuntos, já que a ideia é exatamente avaliar a portabilidade de textos de diferentes domínios: literário e jornalístico. Como já era esperado, os resultados obtidos nessas combinações foram inferiores em comparação às demais. Restringindo a análise aos resultados do classificador AODE, a queda foi de aproximadamente 25% e 20% para A–B e B–A, respectivamente.

Os resultados ruins obtidos com relação a portabilidade podem ser explicados, entre outros fatores, pela diferente distribuição de pronúncias entre os corpora A e B. Assim, este experimento foi repetido equilibrando artificialmente os exemplos de cada pronúncia entre as duas bases de texto. Os resultados são mostrados na Tabela 3. Percebe-se novamente queda de desempenho nas combinações A–B e B–A, ou seja, mesmo quando a mesma distribuição de pronúncias é conservada entre os exemplos de treino e teste, a portabilidade não é garantida. Esse fato mostra que os algoritmos adquirem diferentes (e não permutais) sugestões de classificação de ambas as fontes. Outro ponto relevante foi que o conjunto A+B–A (ou A+B–B) continuou com aproximadamente o mesmo desempenho do conjunto A–A (ou B–B) em todos os algoritmos. Isto é, o conhecimento obtido a partir de um único corpus quase abrange o conhecimento de combinar ambos os corpora.

Tabela 3. Experimento com a mesma distribuição dos HHs entre os corpora.

Algoritmo	Acurácia (%)						
	A+B—A+B	A+B—A	A+B—B	A—A	B—B	A—B	B—A
ZeroR	40,18	38,69	38,62	38,04	38,10	50,00	50,00
NaiveBayes	92,41	92,11	92,30	92,05	91,67	85,71	86,16
AODE	88,62	89,29	89,29	88,75	88,17	80,80	83,04
J48	87,50	86,76	87,05	86,07	86,31	75,00	78,12
RandomForest	69,87	70,54	71,21	73,12	75,00	73,21	74,78

4.2. Segundo Experimento

O primeiro experimento mostrou que os classificadores treinados com o corpus A não funcionaram bem com o corpus B, e vice-versa. Então, esse segundo experimento explora o efeito do processo de incremento (ou *tuning*) na tentativa de tornar os sistemas supervisionados portáveis. Esse processo consiste em adicionar ao conjunto de treino original uma quantidade relativamente pequena de amostras do novo domínio. O tamanho dessa porção supervisionada varia de 10% a 50%, em passos de 10%, sendo que os 50% restantes são reservados para os testes. Os conjuntos treino-teste desse experimento serão denotados por A+%B–B e B+%A–A.

Para analisar a real contribuição do conjunto de treino original na desambiguação de HHs no novo domínio, os valores de acurácia para as combinações %B–B e %A–A também foram calculados. Os resultados desse experimento são apresentados na Figura 1. Cada gráfico contém as curvas X+%Y–Y e %Y–Y, além de três linhas horizontais, que correspondem ao limite inferior, representado pelo rótulo PMF (pronúncia mais frequente dada pelo algoritmo ZeroR), e aos limites superiores X+Y–Y e Y–Y de cada algoritmo. Conforme esperado, a acurácia de todos os métodos aumenta (em direção ao limite superior) à medida que mais amostras do novo domínio são adicionadas ao conjunto de treino. Verificou-se também a degradação provocada pelo corpus de treino original na acurácia dos classificadores, com a curva %Y–Y superando a curva X+%Y–Y em todos os algoritmos. Resumindo, os gráficos mostraram que não é interessante manter os exemplos de treino originais. Ao contrário, uma melhor estratégia, embora desapontadora, é simplesmente usar o corpus incrementado.

5. Conclusões e Trabalhos Futuros

O uso de técnicas de aprendizado de máquina para tratar o problema de ambiguidade entre palavras não é novidade em várias línguas, porém, no que tange ao PB, as referências são escassas. A maioria das pesquisas usam regras linguísticas formuladas com base em análise contextual, contudo, não descrevem claramente a metodologia usada para compor as regras, além de ser uma estratégia de difícil implementação do ponto de visto semântico. Outros estudos baseiam-se de forma limitada em etiquetas gramaticais, onde as colocações destas nas vizinhanças dos HHs determinam a pronúncia adequada.

A abordagem apresentada neste trabalho, baseada em algoritmos inteligentes de classificação supervisionada, visa diminuir a lacuna existente nessa linha de pesquisa para o PB. A ideia é possibilitar uma rápida atualização dos classificadores por meio da adição de novas amostras na base de conhecimento e, claro, a construção de um desambiguador de HHs automático. Um dos principais desafios dessa técnica é a coleta de amostras suficientes para cada palavra de interesse, pois, apesar de existirem textos em grande quantidade acessíveis na Internet, é necessária uma busca específica para obter as amostras, etiquetá-las e alimentar a base de treino. Os resultados iniciais comprovaram a viabilidade do método e apontaram algumas dificuldades com relação a portabilidade de sistemas de desambiguação supervisionada.

Com relação aos trabalhos futuros, pretende-se construir regras linguísticas para tratar algumas categorias de HHs, principalmente onde a informação morfossintática é suficiente para determinar sua tonicidade. O objetivo é ter uma abordagem híbrida. Também é preciso testar outros algoritmos e domínios de aplicação, como redes sociais.

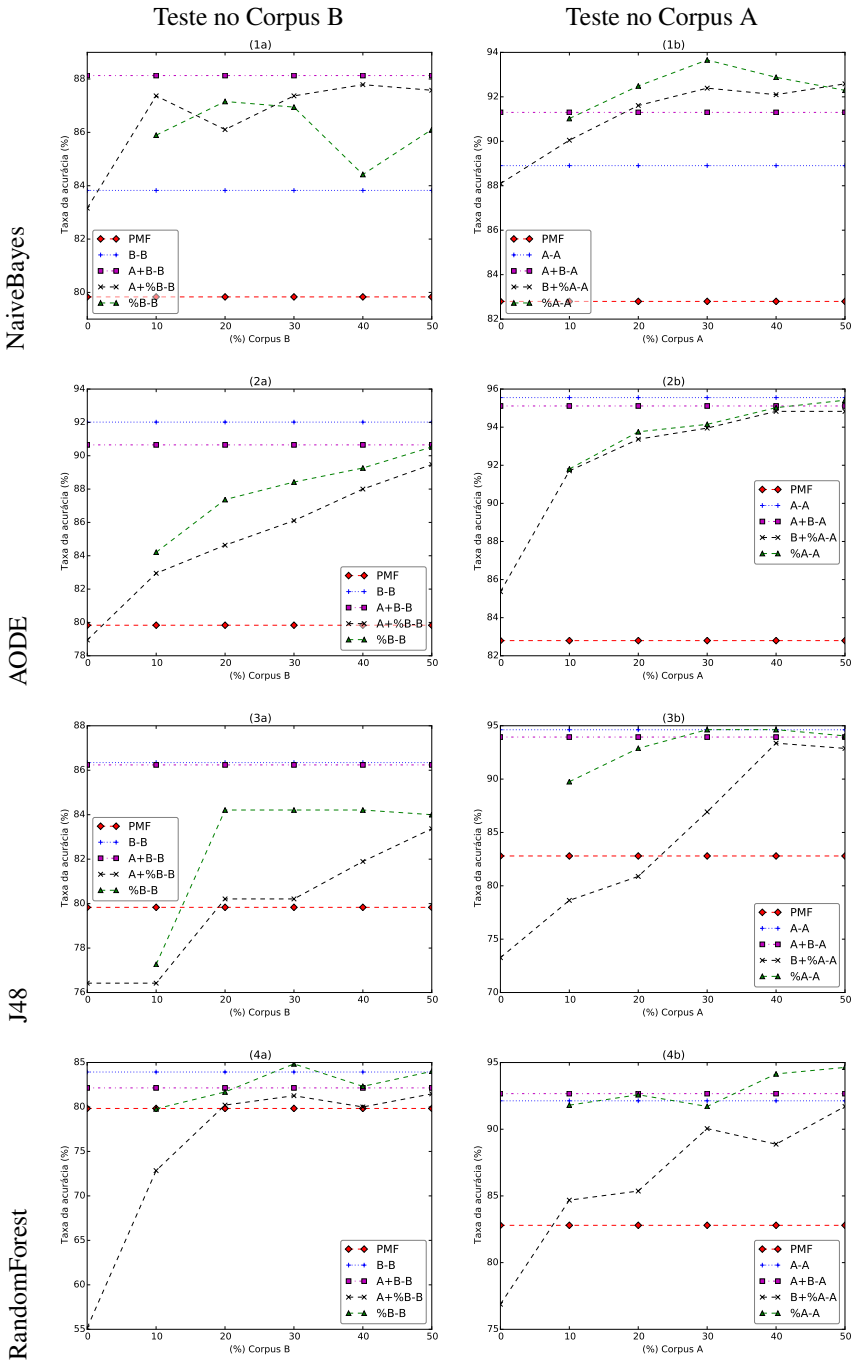

Figura 1. Resultado do experimento de afinação do corpus de treino.

Referências

[ABL 2011] ABL (2011). Espaço Machado de Assis na *Web* da Academia Brasileira de Letras. Disponível em: http://www.machadodeassis.org.br. Acesso em 6 de agosto de 2015.

[Aires et al. 2000] Aires, R., Aluísio, S., Kuhn, D., Andeeta, M., and Oliveira Jr., O. (2000). Combining multiple classifiers to improve part of speech tagging: A case study for brazilian portuguese. In *SBIA 2000 – The Proceeding of the Brazilian AI Symposium*.

[Aluísio et al. 2003] Aluísio, S., Pelizzoni, J., Marchi, R., De Oliveira, L., Manenti, R., and Marquiafável, V. (2003). An account of the challenge of tagging a reference corpus for brazilian portuguese. In *PROPOR'2003 – 6th Workshop on Computational Processing of the Portuguese Language*, pages 110–117, Berlin, Heidelberg. Springer-Verlag.

[Barbosa et al. 2003a] Barbosa, F., Ferrari, L., and Resende Jr., F. G. V. (2003a). A distinção entre homógrafos heterófonos em sistemas de conversão texto-fala. In *Processamento da Linguagem, Cultura e Cognição: Estudos de Linguística Cognitiva*, Braga, Portugal.

[Barbosa et al. 2003b] Barbosa, F., Ferrari, L., and Resende Jr., F. G. V. (2003b). A methodology to analyze homographs for a brazilian portuguese TTS system. In *PROPOR'2003 – 6th Workshop on Computational Processing of the Portuguese Language*, pages 57–61, Berlin, Heidelberg. Springer-Verlag.

[Bielza and Larrañaga 2014] Bielza, C. and Larrañaga, P. (2014). Discrete Bayesian network classifiers: A survey. *ACM Computing Surveys*, 47(1):43. DOI: http://dx.doi.org/10.1145/2576868.

[Brasil 1988] Brasil (1988). Constituição da República Federativa do Brasil de 1988. Disponível em: http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm. Acesso em 6 de agosto de 2015.

[Cardie 1996] Cardie, C. (1996). Embedded machine learning system for natural language processing: A general framework. In *Connectionist, Statistical and Symbolic Approaches to Learning for Natural Language Processing, Lecture Notes in Artificial Inteligence*, pages 315–328. Springer.

[Ferrari et al. 2003] Ferrari, L., Barbosa, F., and Resende Jr., F. G. V. (2003). Construções gramaticais e sistemas de conversão texto-fala: O caso dos homógrafos. In *Proceedings of the International Conference on Cognitive Linguistics*.

[Fonseca and Rosa 2013] Fonseca, E. and Rosa, J. (2013). Mac-Morpho revisited: Towards robust part-of-speech tagging. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, pages 98–107.

[Galves and Faria 2010] Galves, C. and Faria, P. (2010). Tycho Brahe parsed corpus of historical portuguese. Disponível em: http://www.tycho.iel.unicamp.br/~tycho/corpus/en/index.html. Acesso em 6 de agosto de 2015.

[Hall et al. 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1):10–18.

[Han et al. 2011] Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3 edition.

[Màrquez 2000] Màrquez, L. (2000). Machine learning and natural language processing. Technical report, Centre de recerca TALP, Departament de Llenguatges i Sistemes Informàtics, LSI, Universitat Politècnica de Catalunya, UPC, Barcelona.

[Neto et al. 2011] Neto, N., Patrick, C., Klautau, A., and Trancoso, I. (2011). Free tools and resources for brazilian portuguese speech recognition. *Journal of the Brazilian Computer Society*, 17:53–68.

[NILC 2002] NILC (2002). Corpus de extractos de textos electrónicos NILC/Folha de S. Paulo, versão 1.0. Disponível em: http://www.linguateca.pt/CETENFolha. Acesso em 6 de agosto de 2015.

[Portal S. F. 1998] Portal S. F. (1998). Portal São Francisco. Disponível em: http://www.portalsaofrancisco.com.br/. Acesso em 6 de agosto de 2015.

[Ribeiro et al. 2009] Ribeiro, M., Braga, D., Henriques, M., Dias, S., and Rahmel, H. (2009). Resolução de ambiguidades na normalização. In *XXIV Encontro Nacional da Associação Portuguesa de Linguística*, pages 411–426.

[Ribeiro et al. 2003] Ribeiro, R., Oliveira, L., and Trancoso, I. (2003). Using morphosyntactic information in TTS systems: Comparing strategies for european portuguese. In *PROPOR'2003 – 6th Workshop on Computational Processing of the Portuguese Language*, pages 143–150, Berlin, Heidelberg. Springer-Verlag.

[Seara et al. 2001] Seara, I., Kafka, S., Klein, S., and Seara, R. (2001). Considerações sobre os problemas de alternância vocálica das formas verbais do português falado no brasil para aplicação em um sistema de conversão texto-fala. In *SBrT 2001 – XIX Simpósio Brasileiro de Telecomunicações*, Fortaleza, CE.

[Seara et al. 2002] Seara, I., Kafka, S., Klein, S., and Seara, R. (2002). Alternância vocálica das formas verbais e nominais do português brasileiro para aplicação em conversão texto-fala. *Revista da Sociedade Brasileira de Telecomunicações*, 17(1):79–85.

[Shulby et al. 2013] Shulby, C., Mendonça, G., and Marquiafável, V. (2013). Automatic disambiguation of homographic heterophone pairs containing open and closed mid vowels. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, pages 126–137, Fortaleza, CE.

[Silva et al. 2012] Silva, C., Braga, D., and Resende Jr., F. G. V. (2012). A rule-based method for homograph disambiguation in brazilian portuguese text-to-speech systems. *Journal of Communications and Information Systems*, 1(1).

[Witten and Frank 2005] Witten, H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 edition.

[Yarowsky 1997] Yarowsky, D. (1997). Homograph disambiguation in text-to-speech synthesis. In *Progress in Speech Synthesis*, pages 157–172. Springer.