Correção Gramatical da Crase usando Redes Neurais

Andréia Gentil Bonfante
Maria das Graças Volpe Nunes
{andreia, mdgvnune@icmc.sc.usp.br}
NILC-São Carlos
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo
Caixa Postal 668
13560-970 São Carlos -SP

Abstract

This work presents an experiment on the use of neural networks to detect incorrect use of the absence of "crase" in portuguese sentences. Previous expermiment has compared this approach to the symbolist one adopted in the development of ReGra [Bonfante-97]. This paper describes the alterations the authors have proposed in order to improve the previous results and some comparisons between them.

1. Introdução

O uso de técnicas conexionistas aplicadas à correção gramatical automática de textos tem se mostrado de grande valor numa área de implementação tradicionalmente simbolista.

O trabalho com técnicas simbolistas, particularmente o uso de gramáticas descritas como um conjunto de regras, ainda apresenta resultados muito satisfatórios. Tomando como base o revisor gramatical ReGra ([Martins et al.-97],[Nunes-96]), observamos que a eficiência na ferramenta paga um alto preço com relação ao tempo e à quantidade de código que utiliza para uma verificação detalhada de todos os possíveis casos.

Apostando no poder das redes neurais quanto à facilidade de aquisição de conhecimento, já que esta se utiliza somente de um conjunto de exemplos para isso, acreditamos que elas possam ser grandes aliadas das ferramentas tradicionais.

Com a intenção de obter resultados cada vez melhores, realizamos aqui um estudo adicional ao proposto em [Bonfante-97], no qual utilizamos redes neurais para detectar erros com relação ao uso da crase na língua portuguesa do Brasil. Este estudo propõe uma nova seleção e representação do conjunto de exemplos.

A nova seleção e representação dos dados partiu de questionamentos em relação ao tamanho da janela usada, ou seja, se o fato de pegarmos somente uma janela mínima, poderia ter influenciado no resultado.

Nas seções seguintes apresentamos, além dos testes realizados com as redes neurais, uma ferramenta simbolista para a revisão gramatical, bem como citamos alguns casos onde técnicas conexionistas são usadas na resolução de problemas de processamento de linguagem natural.

2. O revisor gramatical ReGra

O ReGra é uma das mais completas ferramentas de revisão gramatical existente para a língua portuguesa falada no Brasil. Por tratar exaustivamente todos os erros relativos à crase, a ferramenta tem servido como parâmetro nos testes realizados em nossos trabalhos com as redes neurais.

O revisor ReGra é composto por três módulos -- mecânico, estatístico e lingüístico -- sendo estes responsáveis pelo tratamento de erros de digitação, por uma análise estatística do texto revisado, e pela verificação quanto aos erros gramaticais propriamente dito, respectivamente. Sendo assim, o que tem relevância aqui é o módulo lingüístico, onde se faz a verificação da crase.

Para a resolução do problema da crase, o módulo lingüístico utiliza-se de duas estratégias: 1.) para a detecção do uso incorreto, a verificação se faz através de regras pontuais, ou seja, verifica-se somente as palavras vizinhas da ocorrência da crase; 2.) para a detecção de ausência, é utilizada a análise sintática fornecida pelo *parser*. Assim, devido às ambigüidades lexicais e de sintaxe encontradas no *parser*, o revisor é mais eficiente na verificação de uso incorreto da crase.

3. Modelos Conexionistas aplicados em Linguagem Natural

A aplicação de modelos conexionistas, particularmente de Redes Neurais, em problemas de Processamento de Linguagem Natural ainda é relativamente pequena quando se trata de revisão gramatical automática. No caso específico de Português, não temos notícia de trabalhos semelhantes. O trabalho de Mendes e Eizirik [Mendes & Eizirik-89] propõe um modelo neuronal probabilístico para resolver alguns problemas simples de ambigüidade, como o caso de múltiplas categorias gramaticais para uma mesma palavra.

Encontramos um outro exemplo, desta vez usando um modelo de Rede Neural Recorrente Simples, em [Servan-Schreiber, Cleeremans & McClelland-89], onde a Rede Neural foi treinada para predizer a próxima letra numa cadeia de caracteres.

Uma outra aplicação que podemos citar como exemplo o uso de Rede Recorrente Simples é na resolução de problema de ambigüidade léxica. Mayberry e Miikkulainen baseiam-se em representações distribuídas de freqüência de contexto para modelar o comportamento humano, com relação ao discernimento automático do conteúdo semântico, diante de tal ambigüidade [Mayberry & Miikkulainen-94].

Um outro trabalho, que nos serviu como base na modelagem e estruturação dos dados, é proposto em [Lawrence et al.-96], onde é feita uma comparação entre vários modelos de Redes Neurais quando aplicados a problemas lingüísticos. Para comparar o desempenho das redes neurais, o autor treinou-as para classificar sentenças em linguagem natural como gramatical ou não gramatical. Dentre os modelos escolhidos estão dois dos modelos mais populares de Redes Neurais: as **Redes Neurais Feed-forward (MLP)** [Rumelhart, Hinton & Williams-86], e as **Redes Neurais Recorrentes Simples (ELMAN)** [Elman-90]. Resultados obtidos com alguns testes apontam 54% de classificações corretas com a rede MLP e 74% com a rede ELMAN.

Trabalhos e estudos mais recentes apontam para o uso de RN recorrentes simples no tratamento de problemas lingüísticos, conforme exemplos apresentados neste capítulo. Uma rede recorrente simples, como é o caso da rede Elman, possui as mesmas características de uma rede treinada com um algoritmo backpropagation (como é o caso da Multilayer Perceptron), já que ela se utiliza também do processo de propagação direta dos valores de ativação, bem como da retropropagação dos erros. A característica que a marca como uma rede recorrente simples é que juntamente com as unidades de entrada, são criadas as unidades de contexto, que possuem ligações diretas com a camada escondida, mas também possuem ligações recorrentes, ou seja, elas recebem de volta o valor da ativação da camada escondida, de modo que este possa ser usado como dado adicional de entrada no próximo passo no tempo de treinamento. Isso, proporciona uma espécie de memória à rede, ajudando-a a se "lembrar" do ultimo padrão que recebeu, tornando-se assim, muito eficiente na tarefa de predição de seqüências, como já dito anteriormente.

Apesar desse fato, temos em contrapartida que são basicamente exemplos de predição, ou seja, a rede é utilizada para produzir determinada seqüência, característica mais marcante das redes recorrentes simples.

Já que no experimento realizado não queríamos ter essa característica, ou seja, não há tarefa de predição do próximo elemento da seqüência e sim, a apresentação desta como um todo, então não iríamos aproveitar a "memória" de uma rede recorrente simples.

Tendo em vista esta informação e também testes realizados com a rede Elman para a mesma aplicação anterior, devido à característica da aplicação proposta aqui neste estudo, optamos pelo uso da rede Multlayer Perceptron para a realização da experiência.

4. O Experimento

A fim de comparar o quanto influenciaria a informação contida no vetor de representação, bem como o tamanho da janela a ser abordada, ou seja, quantas palavras são selecionadas antes da suposta ocorrência de crase, e quantas são selecionadas depois, sugerimos uma forma um pouco mais abrangente (que chamaremos a partir daqui de representação2) do que aquela apresentada na primeira proposta feita pelas autoras em [Bonfante-97] e [Bonfante & Nunes-97] (que chamaremos a partir daqui de representação1).

Para a realização dos testes foi escolhido, para ambas as representações, o modelo de rede neural Multilayer Perceptron [Rumelhart, Hinton & Williams-86], usando o algoritmo *backpropagation* para treinamento. As redes neurais foram construídas e treinadas com a ajuda do Simulador de Redes Neurais **SNNS** (Stuttgart Neural Network Simulator) [Reference Software-95]. Para a obtenção dos padrões de treinamento, ou seja, dos exemplos de ocorrência, foi utilizado um programa que rastreia o *corpus* construído pelos projetistas do ReGra[1]. Assim, foram colhidos tanto exemplos de casos onde a crase está sendo usada, como também contra-exemplos, possibilitando que a rede pudesse ser treinada para reconhecer tanto casos de presença incorreta de crase, como também de ausência. A seguir descrevemos ambas as representações discutidas.

4.1. Descrição da Representação1

A partir da localização da ocorrência, na representação1, é retirada da posição uma janela mínima de sua vizinhança, com somente uma palavra antes e uma depois. Com a janela selecionada, esta é submetida ao classificador sintático (*tagger*) do ReGra, tendo como resultado as classes sintáticas e demais traços dessas palavras. A classificação ocorre entre treze classes principais: substantivo, artigo, preposição, numeral, verbo, pronome, adjetivo, nome próprio, abreviatura, conjunção, advérbio, prefixo e sigla. Estas classes possuem subdivisões tais como gênero, número, regência (no caso verbal e nominal) e transitividade (no caso de verbos), etc.

Para a codificação (representação dos exemplos para serem utilizados pela rede neural), utiliza-se somente as informações previamente consideradas como relevantes: as classes substantivo e adjetivo com informações quanto ao gênero, número e se regem ou não a preposição ‘a’; as classes numeral, nome próprio e abreviatura com informações quanto ao gênero e ao número; a classe pronome com, além de gênero e número, a indicação de ser pronome possesivo ou não; a classe verbo, com informações sobre transitividade, e forma (normal/infinitiva), e ainda sua regência quanto à preposição ‘a’; a classe advérbio, com informações de modo e lugar; e, por fim, classes que só necessitam de indicações de aparecimento que são artigo, preposição, conjunção, prefixo e sigla.

O vetor de entrada para a rede neural é formado da representação das palavras (sendo cada uma representada em 25 posições) e da representação da ocorrência (em 3 posições, sendo que não são tratadas as variações do ‘aquele’), formando-se assim, um vetor de 53 posições. O preenchimento do vetor é feito com indicação de 1 para características presentes na classificação obtida do *tagger* e 0 quando há ausência. Nas Figura 4.1 e 4.2 apresentamos exemplos de representações das palavras e da ocorrência.

Consideramos três saídas diferentes produzidas pela rede: respostas no intervalo ($0 \le r \le 0.4$) são consideradas casos em que a rede classificou os exemplos como incorretos; respostas no intervalo ($0.41 \le r \le 0.6$) são consideradas casos em que a rede ficou em dúvida se os exemplos eram corretos ou não; respostas no intervalo ($0.61 \le r \le 1$) são consideradas casos em que a rede classificou os exemplos como corretos;

[1] O corpus citado é composto por centenas de textos extraídos de jornais, revistas, livros, dissertações, teses e redações de vestibular, constituindo um conjunto de aproximadamente 30 milhões de palavras [Nunes et al.-96b].

O conjunto de treinamento consistiu de 1576 exemplos, dentre eles exemplos negativos e positivos, abrangendo tanto casos de uso incorreto quanto de ausência.

S	A	P	NUM	V	PRON	ADJ	NOM	ABREV	CONJ
0	0	0	0	1	0	0	0	0	0

ADV	PREF	SIGL	F	M	SI	PL	REG	TD	TI
0	0	0	0	0	0	0	1	0	1

N	INF	POSS	MOD	LUG
1	0	0	0	0

Figura 4.1: Codificação de uma palavra (representação1).

C	SI	PL
1	1	0

Figura 4.2: Codificação da ocorrência (representação1).

4.2. Descrição da Representação2

Na representação2 foi feita a opção por ser retirada uma janela de maior abrangência com relação à da representação1, passando a conter três palavras antes e duas depois da ocorrência; a classificação entre as treze classes é mantida, bem como informações quanto a gênero, número, regência e formas do verbo (normal/infinitivo). As informações de transitividade dos verbos e tipos de advérbios foram desconsideradas, sendo adicionadas informações quanto ao tipo de pronomes (demonstrativos, possessivos, indefinidos, relativos, pessoais de tratamento e interrogativos).

Uma representação mais concisa diminuiu de vinte e cinco para quatorze posições por palavra no vetor. A ocorrência também recebeu uma nova representação.

O preenchimento do vetor de entrada da rede também sofreu mudanças quanto à forma. Ao invés de termos uma posição para representar cada característica, temos agora as posições recebendo combinações de zeros e uns, sendo cada combinação representando uma característica. Assim, as quatro posições iniciais representam as classes gramaticais, as duas seguintes o gênero, e assim por diante, conforme mostrado na Figura 4.3. A ocorrência também é representada usando-se combinações (inclusive para as variações do 'aquele'), conforme apresentado na Figura 4.4.

0	0	0	1	S
0	0	1	0	ADJ
0	0	1	1	P
0	1	0	0	NUM
0	1	0	1	V
0	1	1	0	PRON
0	1	1	1	A
1	0	0	0	NOM
1	0	0	1	ABREV
1	0	1	0	CONJ
1	0	1	1	ADV
1	1	0	0	PREF
1	1	0	1	SIGL

0	1	F
1	0	M

0	1	SI
1	0	PL

	REG

0	1	N
1	0	INF

0	0	1	demons
0	1	0	poss
0	1	1	indef
1	0	0	relat
1	0	1	trat
1	1	0	inter

Figura 4.3: Codificação de uma palavra (representação2).

A análise da saída da rede é feita da mesma forma que a descrita na representação1. O conjunto de treinamento consiste de 1500 exemplos envolvendo sentenças corretas e incorretas gramaticalmente, em iguais proporções de aparecimento. As sentenças também foram retiradas do *corpus* citado acima.

Os textos do *corpus* onde foram retirados consistem de textos de jornais, que, supostamente estão escritos de forma correta. Os exemplos foram trabalhados e produziu-se a partir daí contra-exemplos.

0	0	a sem crase
0	1	a com crase
1	0	aq com crase
1	1	aq sem crase

0	1	SI
1	0	PL

Figura 4.4: Codificação da crase (representação2).

5. Testes Realizados

Os testes foram efetuados com 170 sentenças também retiradas do *corpus*. Segundo uma análise geral envolvendo as respostas das redes para todos o exemplos contidos no conjunto de testes, percebe-se que as duas apresentam uma variação pequena em relação às classificações dos exemplos, com uma pequena vantagem da representação2 sobre a representação1, conforme podemos observar na Tabela1, estando especificada em porcentagens a quantidade de casos em que houve acertos, erros ou então onde houve dúvidas com relação às respostas das redes.

Tabela1. Testes gerais realizados

Repres.	Acertos	Erros	'Dúvidas'
1	60%	28%	12%
2	64%	23%	13%

Uma segunda contagem foi considerada com um subconjunto de testes, mostrado nas Tabelas 2 e 3. Consistiu na análise separada de exemplos onde encontramos casos de regência nominal (56 exemplos) e ainda de exemplos onde encontramos casos de regência verbal (40 exemplos). Considerando os casos de regência nominal, observamos um aumento significativo de acertos da representação2 com relação à representação1. Os erros também tiveram uma diferença significativa. Já quanto às duvidas, a representação2 apresentou um índice bem mais alto que a anterior.

Tabela2. Testes com Regência Nominal

Repres.	Acertos	Erros	'Dúvidas'
1	57%	34%	9%
2	68%	16%	16%

Para os casos de regência verbal obtivemos também um comportamento melhor, ainda que pequeno, da representação2.

Tabela3. Testes com Regência Verbal

Repres.	Acertos	Erros	'Dúvidas'
1	47.5%	45%	7.5%
2	50.0%	47.5%	2.5%

Nas Tabelas 4 e 5 apresentamos, respectivamente, resultados analisando-se separadamente respostas das redes para exemplos onde aparece a crase (para detecção de uso incorreto) e respostas para exemplos onde não aparece de crase (para detecção de ausência). No caso de detecção de uso incorreto, a representação1 obteve um resultado levemente favorável ao da representação2 com relação à quantidade de acertos e dúvidas; já nos erros, ocorreu o contrário.

Nos casos onde não há uso de crase, obtivemos uma significante melhora da representação2, sendo que ela consegue detectar mais eficientemente os casos de ausência de crase.

Tabela4. Detecção de uso incorreto

Repres.	Acertos	Erros	'Dúvidas'
1	67%	19%	14%
2	63%	17%	20%

Tabela5. Detecção de ausência

Repres.	Acertos	Erros	'Dúvidas'
1	45%	33%	22%
2	65%	26%	12%

De acordo com o que foi citado acima, pudemos observar variações no comportamento das duas redes. A representação 2, apesar de ser mais abrangente em sua abordagem (uma janela maior da vizinhança da ocorrência), não consegue ter resultados significativamente satisfatórios com relação à quantidade de acertos produzidos. Ainda assim, em alguns casos, a representação2 consegue superar em muito a representação1. Casos como detecção de ausência de crase, um problema que ainda está em aberto pelo ReGra. Talvez esteja aí a eficiência esperada. Uma outra característica observada no comportamento geral da representação2 é a diminuição significativa da quantidade de erros, sendo amenizado pelo aumento das respostas "duvidosas" da rede. Em última análise, é mais significativo para nós que a rede fique em dúvida com relação à determinda apresentação, pois assim podemos considerá-la e tratá-la de uma forma especial. Ou seja, tendo como objetivo indicar ao usuário da ferramenta trechos no texto analisado onde exista uma estrutura não correta gramaticalmente para que este possa corrigí-la, é mais satisfatório não apresentarmos trechos duvidosos, do que apresentarmos trechos supostamente corretos para serem corrigidos.

Vale a pena ainda citar alguns dados obtidos com a análise do conjunto de testes pelo ReGra: para os casos onde temos o uso de crase, o comportamento do revisor é de 100% de acerto; para casos onde não há uso, sendo pois, verificada a ausência de crase, observamos que o revisor pede interferência do usuário em 27.5% do total de exemplos a ele apresentados, isto é, o revisor não tem certeza absoluta do erro, apontando apenas possíveis erros dependentes da avaliação do usuário. Ao contrário de uma rede neural, onde há uma obrigatoriedade de apresentação de resultado a todos os exemplos propostos, não temos aqui como verificar se a ferramenta considerou os casos restantes como corretos gramaticalmente ou se simplesmente ignorou-os por não estarem sendo tratados.

Tendo em vista esses resultados, podemos chegar a conclusões ainda positivas com relação ao uso de redes neurais em aplicações desse tipo. O revisor ReGra consegue resultados tão satisfatórios em casos de deteção de uso incorreto devido ao fato de que, como possui características simbolistas, pode ser adicionado à ferramenta um número muito grande e exaustivo de regras pontuais que abrangem praticamente todos os casos a serem tratados. Essa característica não temos na rede neural, onde a única fonte de informação são os exemplos a ela apresentados. Daí entra em cena a capacidade da rede de abstrair e generalizar esses exemplos. Já em casos de tratamento de ausência, o revisor ReGra tem uma maior dificuldade por não se tratar simplesmente uma análise de regras pontuais e sim, de uma análise sintática sobre toda a frase, complicando o processo. A rede neural, por sua vez, consegue a partir dos exemplos, retirar informações satisfatórias sobre como lidar com esses casos.

Por fim, apresentamos na Tabela6 alguns exemplos de janelas e o comportamento das redes treinadas. A letra 'a' indica que a rede acertou na resposta e a letra 'e' que a rede errou na resposta.

Tabela6. Exemplos de comportamento das redes

Exemplo	Repr1	Repr2
... respostas dos estudantes às ações do... (correto)	e	a
... comentário à respeito de .... (incorreto)	e	a
... a adaptação as necessidades do ... (incorreto)	a	e
... desigualdade de acesso a informação ... (incorreto)	a	e
... relacionada diretamente à crença segundo ... (correto)	e	a
... a informática poderá à primeira vista ... (correto)	a	e

6. Conclusões

Este artigo apresentou trabalho relativo ao uso de redes neurais na verificação do uso de crase em sentenças da língua portuguesa. Trata-se de uma extensão de trabalho previamente realizado, cujos resultados puderam ser contrastados entre si. Conforme pudemos observar, os resultados obtidos, quando comparados aos anteriores, apresentam sinais de melhora, ainda que abaixo do que as autoras esperavam. Outros modelos de redes têm sido aplicados a PLN e nossa intenção é realizar outros experimentos a fim de concluir sobre os fatores que de fato podem influenciar nos resultados obtidos. Para longo prazo, pretende-se construir ferramentas híbridas de verificação gramatical para o português, combinando recursos simbólicos com conexionistas.

7. Referências Bibliográficas

[Bonfante-97] Bonfante, A.G. Uso de Redes Neurais para Correção Gramatical do Português: Um Estudo de Caso. Tese de Mestrado, ICMSC - Departamento de Computação, Março,1997.

[Bonfante & Nunes-97] Bonfante, A.G.; Nunes, M.G.V. *Redes Neurais e Correção Gramatical do Português: um estudo de caso.* III Congresso Brasileiro de Redes Neurais. Florianópolis, Julho 1997, pp. 262-267.

[Elman-90] Elman, J.L. Finding structure in time. *Cognitive Science Society*, vol.14, pages 179-211, 1990.

[Lawrence et al.-96] Lawrence, S.; Fong, S.; Giles, L. Natural Language Grammatical Inference: A Comparison of Recurrent Neural Networks and Machine Learning Methods. In: *Symbolic, Connectionist, and Statistical Approaches to Learning for Natural Language Processing. Lecture Notes in AI series.* Springer Verlag, Berlin, 1996.

[Martins et al.-97] Martins, R.T.; Hasegawa; Nunes, M.G.V.; Montilha, G.; Oliveira Jr., O.N.; Linguistic issues in the development of ReGra: a Grammar Checker for Brazilian Portuguese. *Natural Language Engineering.* Aceito em junho de 1997.

[Mayberry & Miikkulainen-94] Mayberry, M.R.; Miikkulainen, R. Lexical Disambiguation Based on Distribute Representations of Context Frequency. In*: Procedings of the 16th Annual Conference of the Cognitive Science Society*, 1994.

[Mendes & Eizirik-89] Mendes, S.B.T.; Eizirik, L.M.R. Um Modelo Neuronal Probabilístico para Solução da Ambigüidade do Português. *Revista Brasileira de Computação* Jul/Set vol.5 pp.45-51, COPPE, Rio de Janeiro, 1989.

[Nunes et al.-96] Nunes, M.G.V.;Ghiraldelo, C.M.; Montilha, G.; Turine, M.A.S.; Oliveira, M.C.F.; Hasegawa, R.; Martins, R.T.; Oliveira Jr.,O.N. Desenvolvimento de um Sistema de Revisão Gramatical Automática para o Português do Brasil. In: *II Encontro para o Processamento Computacional do Português Escrito e Falado*, Curitiba, PR, 1996.

[Reference Software-95] SNNS (Stuttgart Neural Network Software) V4.1. User's Guide, 1995.

[Rumelhart, Hinton & Williams-86] Rumelhart, D.E.; Hinton, G.E., & Williams, R.J. Learning Internal Representations by Error Propagation. In D.E. Rumelhart & J.L. McClelland(Eds.), *Parallel Distributed Processing: Explorations in the Microestructure of Cognition* (Vol.1, pp.318-362). Cambridge, MA: MIT Press, 1986.

[Servan-Schreiber et al.-89] Servan-Schreiber, D.; Cleeremans, A.; McClelland, J.L. Learning Sequential Structure in Simple Recurrent Networks. In: *Advances in Neural Information Processing Systems.* San Mateo, CA, 1989.