

# Uma arquitetura híbrida LSTM-CNN para reconhecimento de entidades nomeadas em textos naturais em língua portuguesa

Carlos A. E. M. Júnior<sup>1</sup>, Luciano A. Barbosa<sup>2</sup>, Hendrik T. Macedo<sup>1</sup>,

<sup>1</sup>Departamento de Computação  
Universidade Federal de Sergipe (UFS)  
São Cristóvão, SE – Brazil

<sup>2</sup>Centro de Informática  
Universidade Federal de Pernambuco (UFPE)  
Recife, PE – Brazil

{juniorcaemj, lucianobarbosa}@gmail.com, hendrik@dcomp.ufs.br

**Resumo.** *Existem modelos de classificadores que realizam bem a tarefa do reconhecimento de entidades nomeadas em texto a partir de grandes esforços em definir muitas características baseando-se em conhecimento especializado. Neste trabalho é proposta uma arquitetura de Rede Neural Profunda que utiliza uma rede LSTM para detectar entidades nomeadas sobre vários corpora da língua portuguesa do Brasil. Além disso, são usados word embeddings pré-treinados e uma rede CNN para extração automática de features das palavras para geração de vetores de char embeddings. No trabalho são definidos cenários para experimentação onde a arquitetura proposta é comparada com uma arquitetura que também utiliza word e char embeddings, obtendo melhor desempenho utilizando-se apenas word embeddings, e em outro cenário a arquitetura proposta é comparada com um classificador CRF, obtendo melhor desempenho.*

**Abstract.** *Currently there are many machine learning models that perform well Named entity recognition task, by defining features based on well-knowledge expertise. This paper proposes a deep neural network architecture that uses an LSTM network to perform named entity recognition on various corpora in Brazilian Portuguese. Moreover, the proposed solution used pre-trained word embeddings and a CNN network for automatic feature extraction of words and to generate char embeddings vectors. In this work, scenarios were defined in which the proposed architecture is compared to the one which also uses word and char embeddings, obtaining better performance using only word embeddings, and in the other scenario, the architecture is compared with a CRF classifier, getting better performance.*

## 1. Introdução

O reconhecimento de entidades nomeadas é uma atividade importante na área de Extração de Informação para a detecção e classificação de informações presentes em um texto, como por exemplo, pessoas, locais, organizações, datas, eventos. Um bom classificador de entidades nomeadas permite a criação de diversas bases de informação estruturada,

úteis para diversas aplicações, ou ainda sistemas de Q&A (Perguntas e Respostas), como por exemplo o IBM Watson [Murdock 2015].

Tipicamente, *features* peculiares a uma determinada entidade nomeada são pré-definidas e através de uma base de texto natural, conhecida como *corpus*, é treinado um classificador de sequência para rotular as entidades adequadamente. Em [Zaccara 2012], é desenvolvida uma plataforma de reconhecimento de entidades nomeadas para notícias de esporte onde foram usadas várias técnicas de classificação, como por exemplo, um classificador MaxEnt [Borthwick 1999] e máquinas de vetores de suporte [Vapnik 1998]. Em [Júnior et al. 2014], foi utilizado um classificador NaiveBayes [Friedman et al. 1997] para detectar entidades nomeadas de pessoas, organizações e locais para o sistema CA<sup>2</sup>JU Ilustrado, que auxilia pessoas com paralisia cerebral severa a se comunicar e a compreender palavras-chave da língua portuguesa. Em [Júnior et al. 2015], foi utilizado um classificador CRF [Lafferty et al. 2001] com a definição manual de várias *features* específicas para avaliar a classificação sobre o *corpus* Paramopama. O modelo obteve bom desempenho de classificação nos cenários de experimentação testados.

Nos últimos anos têm se popularizado o uso de Redes Neurais Profundas (Deep Learning) em diversas atividades de reconhecimento automático de padrões. No reconhecimento de entidades nomeadas, redes profundas podem aprender relações morfológicas, sintáticas e semânticas sobre uma ou mais palavras presentes na sentença que permitiriam identificar mais precisamente um tipo de entidade em particular [LeCun et al. 2015]. [Collobert et al. 2011] propôs uma rede neural profunda onde são utilizadas Redes Neurais Convolucionais (CNN) para extração de *features* sintáticas e semânticas de janelas de contexto de palavras em uma sentença. O trabalho foi capaz de gerar um modelo que conseguiu atingir o estado da arte para o reconhecimento de entidades nomeadas para a língua inglesa. Já no trabalho [Chiu and Nichols 2015] é proposta uma arquitetura híbrida, combinando uma rede recorrente Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber 1997] bidirecional com uma rede CNN. A rede utiliza uma CNN para extração de *features* léxicos de janelas de contexto das palavras do *corpus* e utiliza uma abordagem não-supervisionada para treinar *features* de palavras sobre o *corpus* da agência de notícia Reuters. O resultado do trabalho se aproximou do estado da arte obtido no CoNLL-2003 [Tjong Kim Sang and De Meulder 2003] e superou o estado da arte dos resultados obtidos sobre o *corpus* OntoNotes. [dos Santos and Guimaraes 2015] utilizou a combinação de *word embeddings* e *char embeddings* [Turian et al. 2010], para classificação de entidades nomeadas para o português do Brasil e para o espanhol. O trabalho utiliza uma abordagem não-supervisionada para aprendizagem dos *word embeddings* e uma rede CNN para geração dos *char embeddings*. Os *embeddings* são então passados à camadas densas para extrair mais níveis de representações. Para o português, a utilização dos *char embeddings* e *word embeddings* obteve resultados que superaram o estado da arte do evento do PrimeiroHAREM [Santos and Cardoso 2007].

O objetivo desse artigo é propor e avaliar uma arquitetura de rede neural profunda para reconhecimento de entidades nomeadas. São utilizados modelos com *word embeddings* e *char embeddings* para extração automática de diversos *features* sobre os dados. Os *word embeddings* utilizados são gerados através de uma abordagem não-supervisionada e os *char embeddings* serão extraídos através de uma rede CNN aplicada sobre as palavras. Além disso, foi definido um cenário de experimentação com o objetivo

de comparar a arquitetura proposta nesse trabalho com a arquitetura proposta por [dos Santos and Guimaraes 2015] e outro cenário para comparar a arquitetura proposta com o classificador CRF utilizado em [Júnior et al. 2014], de forma a comparar o desempenho de uma rede neural profunda com um classificador tradicional.

## 2. Arquitetura Proposta

Nesta seção será descrita a arquitetura da rede utilizada, assim como suas características, os hiper-parâmetros e a arquitetura da camada de entrada. A arquitetura completa da rede pode ser visualizada na figura 1.

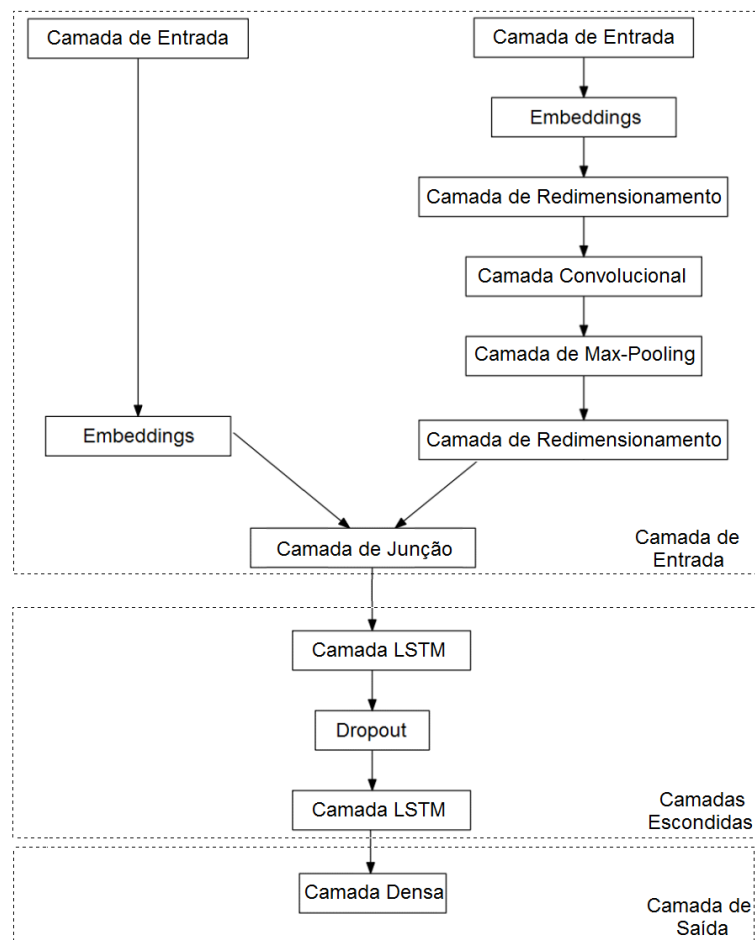
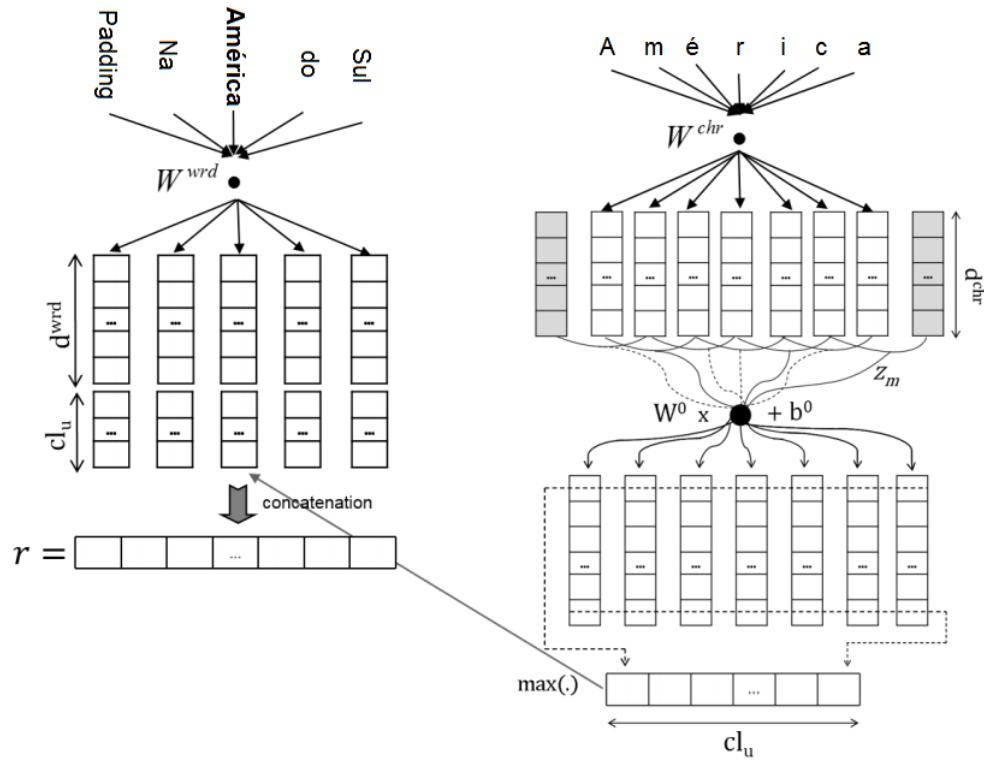


Figure 1. Arquitetura da rede proposta.

### 2.1. Camada de Entrada

A rede neural proposta possui duas entradas (figura 2). Por uma entrada (à esquerda), são passadas as janelas de contexto das palavras das sentenças (com a palavra no centro e as demais palavras ao lado de acordo com o tamanho da janela) e cada palavra é individualmente codificada como um vetor de *word embedding* de dimensão  $d^{word}$ . Os *word-embeddings* utilizados no trabalho foram produzidos em Al-Rfou (2015). Pela outra entrada da rede (à direita), são passadas as janelas de contexto dos caracteres das palavras presentes na janela de contexto. Assim, cada caracter é primeiro convertido em um vetor

de *embeddings*, de dimensão  $d^{chr}$  e inicializados conforme descrito na seção 2.2. Em seguida serão passados a uma camada convolucional que irá extrair *features* de janelas de contexto de caracteres das palavras, e uma camada de *max-pooling* que irá propagar apenas os *features* mais relevantes detectados na convolução. Ao fim, é gerado um vetor  $cl_u$  da saída da CNN que é concatenado com o vetor  $d^{wrd}$ , gerando o vetor  $r$  que será então passado para as camadas LSTM da rede para extração de *features* da sentença.



**Figure 2. Representação da entrada da rede. Fonte: adaptação de Santos (2015)**

Uma variação da arquitetura contendo apenas os *word embeddings* na camada de entrada (lado esquerdo) foi também definida.

## 2.2. Inicialização dos pesos das camadas

O método utilizado para inicialização dos pesos em todas as camadas da rede foi através da distribuição uniforme proposta por [Glorot and Bengio 2010], onde se constatou que os pesos inicializados dessa forma são melhor propagados em redes profundas sem diminuição ou explosão dos sinais no caso dos pesos serem muito pequenos ou muito grandes. A distribuição pode ser calculada através da equação 1, onde  $n_{in}$  é o número de neurônios de entrada da camada e  $n_{out}$  é o número de neurônios de saída da camada.

$$W = Uniform \left[ -\frac{\sqrt{6}}{\sqrt{n_{in} + n_{out}}}, \frac{\sqrt{6}}{\sqrt{n_{in} + n_{out}}} \right] \quad (1)$$

## 2.3. Funções de Ativação

Foram usados dois tipos de funções de ativação na rede. Para as camadas escondidas, foi utilizada a função não-linear tangente hiperbólica, equação 2, que obteve melhores

resultados para as redes LSTM. E para a camada de saída, foi utilizada a função softmax para que pudesse ser gerada uma distribuição de probabilidade entre as possíveis opções de entidades nomeadas, equação 3.

$$\tanh(t) = \frac{e^t - e^{-t}}{e^t + e^{-t}} \quad (2)$$

$$\text{softmax}_i(a) = \frac{\exp a_i}{\sum \exp a_i} \quad (3)$$

## 2.4. Função de Custo

A função de custo escolhida que a rede irá minimizar durante a fase de treinamento é a *Categorical Cross Entropy*, equação 4, pois calcula a entropia cruzada entre a distribuição correta de probabilidade  $p$ , codificada em um vetor one-hot, e a distribuição de probabilidades da saída da rede neural  $q$ , gerada pela função softmax.

$$\text{CrossEntropy}(p, q) = - \sum_x p(x) \log q(x). \quad (4)$$

## 2.5. Método de Atualização dos Pesos

O método de atualização escolhido foi o *Adaptative Gradient(Adagrad)* [Duchi et al. 2011] que é uma variação do *Stochastic Gradient Descent(SGD)*, mas que varia a taxa de aprendizagem da rede de acordo com os dados. Ou seja, quanto mais esparsos são os parâmetros, mais a taxa de aprendizagem aumenta e quanto menos esparsos, mais a taxa de aprendizagem diminui. O Adagrad é um método que tem rápida convergência quando os dados são esparsos e os parâmetros esparsos possuem mais informação, como em aplicações de PLN. O Adagrad é mostrado na equação 5, onde  $\eta$  é a taxa de aprendizagem inicial,  $w_j$  representa o vetor de pesos a ser atualizado e  $G_{j,j}$  é o fator pelo qual aumentará ou diminuirá a taxa de aprendizagem na iteração  $j$ .

$$G_{j,j} = \sum_{\tau=1}^t g_{\tau,j}^2, \quad \text{onde } g_{\tau} = \nabla Q_i(w) \quad (5)$$

$$w_j := w_j - \frac{\eta}{\sqrt{G_{j,j}}} g_j. \quad (6)$$

## 2.6. Demais Hiper-parâmetros

Muitos dos hiper-parâmetros definidos para a rede foram escolhidos baseando-se no trabalho de Santos (2015). Os demais foram definidos baseando-se nos trabalhos de Bengio (2012) e Mikolov (2012), que indicam boas práticas na escolha dos parâmetros e também através de testes realizados de forma empírica com o objetivo de obter o melhor resultado para Medida-F. Nesses testes, um parâmetro era variado enquanto os demais permaneciam fixos; ao longo do treinamento o modelo era avaliado.

Na tabela 1 são relacionados todos os hiper-parâmetros definidos para a rede que será utilizada no experimento, assim como seus respectivos valores.

Table 1. Hiper-parâmetros da rede

Parâmetro	Valor
Tamanho da janela de contexto de palavras	5
Tamanho da janela de contexto de caracteres	5
Dimensão dos <i>Word Embeddings</i>	64
Dimensão dos <i>Char Embeddings</i>	20
Número de camadas escondidas	2
Quantidade de unidades convolucionais	10
Quantidade de unidades LSTM	420
Tamanho do minibatch de treinamento	20
Número de épocas	50
Dropout	50%

### 3. Experimentos

Nesta seção serão descritos os experimentos realizados para calcular a eficiência de classificação de diversos modelos. Serão descritos os *corpora* utilizados nos experimentos, os modelos que serão comparados com a arquitetura proposta, as métricas utilizadas para medir a eficiência de classificação e os cenários de experimentação criados.

Neste experimento foi utilizado o framework *Keras* [Chollet 2015] para a implementação da rede proposta e os testes foram realizados em uma máquina Intel Core 2 Quad, 4GB RAM, com GPU Geforce GTX 650.

#### 3.1. Descrição dos *corpora*

Neste experimento foram utilizados os seguintes *corpora*: Primeiro HAREM [Santos and Cardoso 2007], MiniHAREM [Santos and Cardoso 2007], Paramopama [Júnior et al. 2015], 10% finais do corpus WikiNER [Nothman et al. 2012] e 10% finais do corpus SegundoHAREM [Santos and Cardoso 2007]. No PrimeiroHAREM e no miniHAREM serão avaliados cinco tipos de entidades nomeadas: Pessoa, Local, Organização, Tempo e Valor. Já nos *corpora* Paramopama, WikiNER e no SegundoHAREM, serão avaliados quatro tipos de entidades nomeadas: Pessoa, Local, Organização e Tempo. Detalhes dos três *corpora* utilizados no experimento são mostrados na tabela 2.

Table 2. Detalhes sobre os *corpora* utilizados no experimento

<i>Corpus</i>	Sentenças	Tokens	Tipo
PrimeiroHAREM	4.749	93.125	Treinamento
MiniHAREM	3.393	62.914	Teste
Paramopama	12.500	310.000	Treinamento
WikiNER	5.855	149.613	Teste (Test Set 1)
SegundoHAREM (10%)	383	10.417	Teste (Test Set 2)

### 3.2. Baselines

As arquiteturas propostas (nomeadas **ParamopamaCWNN** com *word embeddings* e *char embeddings* e **ParamopamaWNN** apenas com *word embeddings*) são comparadas com dois modelos. O primeiro modelo é uma arquitetura de rede neural profunda, proposta por Santos (2015) e baseada na rede de Collobert (2011), que utiliza *word embeddings* e *char embeddings*, esses últimos sendo extraídos através de uma rede convolucional, melhorando o desempenho do modelo treinado apenas com *word embeddings*. Nesta comparação, são utilizados os resultados obtidos com *word embeddings* e *char embeddings* (referenciada como **CharWNN**) e também apenas com *word embeddings* (referenciada como **WNN**). Os hiper-parâmetros do modelo são os descritos em [dos Santos and Guimaraes 2015].

O segundo modelo a ser comparado é uma arquitetura de classificador CRF (Conditional Random Fields) (referenciado como **ParamopamaCRF**, para comparação) proposto por Júnior (2015), onde é utilizado um CRF de ordem dois para avaliar o resultado sobre o corpus Paramopama [Júnior et al. 2015]. Neste trabalho existem dois conjuntos de teste. Portanto, a arquitetura proposta será comparada em ambos os conjuntos de teste. Os *features* definidos nesse trabalho estão descritos em [Júnior et al. 2015].

### 3.3. Métricas

As métricas utilizadas nos experimentos foram as métricas tradicionais da literatura para classificação de entidades nomeadas: *Precision*, *Recall* e *F-measure*, equações 7, 8 e 9, onde *TP* são os verdadeiros-positivos (acertou em classificar a palavra em uma entidade nomeada), *FP* são os falsos-positivos (errou em classificar a palavra em uma entidade nomeada) e *FN* são os falsos-negativos (errou em não classificar a palavra em uma entidade nomeada).

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

$$Recall = \frac{TP}{TP + FN} \quad (8)$$

$$F-measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (9)$$

Como o F-measure é a média harmônica entre Precision e Recall, é necessário atribuir pesos para os dois parâmetros e, por entender que ambos são igualmente relevantes para a qualidade do modelo, foi definido o mesmo peso para ambos.

### 3.4. Cenários de Experimentação

Neste experimento foram utilizados dois cenários. No primeiro cenário, nomeado HAREM I, os resultados de F-measure do modelo são comparados com os resultados

do modelo proposto no trabalho de Santos (2015). O *corpus* utilizado neste primeiro cenário foi o PrimeiroHAREM para treinamento e o *corpus* MiniHAREM para teste. Os modelos que serão comparados estão descritos na seção 3.2.

No segundo cenário de experimentação, nomeado Paramopama, os resultados de F-measure do modelo são comparados com os resultados obtidos por Júnior (2015), através de um classificador CRF. Os *corpora* utilizados nesse teste são os mesmos daquele trabalho, sendo o *corpus* Paramopama para treinamento e os extratos dos *corpora* SegundoHAREM e WikiNER para teste. Os modelos que serão comparados estão descritos na seção 3.2.

O objetivo do primeiro cenário é comparar o modelo desse trabalho com outro modelo utilizando os mesmos *corpora*. No segundo cenário, o objetivo é mostrar a eficiência de redes neurais profundas na tarefa de classificação de entidades nomeadas em comparação com os modelos de classificação tradicionais como o CRF.

## 4. Resultados e discussão

Os resultados dos experimentos do Cenário HAREM I e do Cenário Paramopama são descritos a seguir e estão mostrados nas tabelas 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14 e 15.

### 4.1. Cenário HAREM I

Na tabela 3 são mostrados os resultados dos três modelos testados. Nesse teste o modelo com *word embeddings* e *char embeddings* obteve o melhor resultado, seguido de perto pelo modelo com apenas *word embeddings*. Na tabela 4 vemos que o modelo ParamopamaCWNN foi ligeiramente superior por 0.12 pontos percentuais de F-measure do modelo proposto no trabalho de Santos (2015). Na tabela 5 vemos que o modelo ParamopamaWNN obteve 7.72 pontos percentuais sobre o modelo proposto por Santos (2015), o que sugere que o modelo com as camadas LSTM obtiveram melhor desempenho na tarefa de classificação de entidades nomeadas do que o modelo MLP tradicional quando não se utiliza *char embeddings* no modelo.

**Table 3. Resultados dos modelos testados no cenário HAREM I**

Modelo	Precision	Recall	F-measure
ParamopamaCWNN	<b>75.13%</b>	68.38%	<b>71.35%</b>
ParamopamaWNN	73.68%	<b>69.26%</b>	71.22%

**Table 4. Comparação com resultados do trabalho [dos Santos and Guimaraes 2015]**

Modelo	Precision	Recall	F-measure
ParamopamaCWNN	<b>75.13%</b>	68.38%	<b>71.35%</b>
CharWNN	73.98%	<b>68.68%</b>	71.23%

Nas tabelas 6 e 7 são mostrados os resultados por entidade do modelo com ParamopamaCWNN e ParamopamaWNN. Podemos perceber que o modelo ParamopamaCWNN obteve melhor desempenho na classificação das entidades Valor e



**Table 5. Comparação com resultados do trabalho [dos Santos and Guimaraes 2015]**

<b>Modelo</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
ParamopamaWNN	<b>73.68%</b>	<b>69.26%</b>	<b>71.22%</b>
WNN	68.91%	58.77%	63.44%

Organização em relação ao ParamopamaWNN, enquanto o ParamopamaWNN foi melhor nas demais entidades. A melhoria nas entidades do tipo Valor e Organização mostra a importância de *features* relacionados com caractere para entidades desse tipo.

**Table 6. Resultados do modelo ParamopamaCWNN por entidade nomeada**

<b>Entidade</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
Pessoa	75.48%	76.44%	75.95%
Local	70.36%	66.86%	68.57%
Organização	59.12%	50.07%	54.22%
Tempo	58.36%	52.78%	55.43%
Valor	89.81%	65.59%	75.81%

**Table 7. Resultados do modelo ParamopamaWNN por entidade nomeada**

<b>Entidade</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
Pessoa	79.66%	74.26%	76.87%
Local	71.92%	65.85%	68.75%
Organização	55.03%	49.42%	52.07%
Tempo	57.69%	64.19%	60.77%
Valor	80.14%	63.48%	70.85%

#### 4.2. Cenário Paramopama

No cenário Paramopama temos os resultados do Test Set 1 nas tabelas 8, 9, 10 e 11. Na tabela 8 vemos que o modelo que obteve melhor desempenho foi o modelo ParamopamaWNN, obtendo 88.08% de F-measure. Na tabela 9 a comparação entre os modelos propostos e o ParamopamaCRF, mostrou uma diferença de pouco mais de 8 pontos percentuais, comprovando a eficiência de modelos de Deep Learning em comparação com classificadores tradicionais. O que pode justificar a superioridade do ParamopamaWNN em relação ao ParamopamaCWNN pode ser a ausência da entidade Valor que, como mostrado na seção 4.1, conseguiu uma melhora significativa com a adição dos *char embeddings*. Outro fato que ocorreu, como mostrado nas tabelas 10 e 11, foi a diminuição acentuada do *Recall* da entidade Tempo assim como ocorreu no cenário HAREM I, por motivos ainda desconhecidos.

Nas tabelas 12, 13 e 14 e 15, podemos perceber que os resultados foram similares aos obtidos no test set 1, onde o modelo com *word embeddings* apenas obteve um desempenho superior entre os modelos testados. A tabela 13 mostra uma diferença de pouco mais de 4 pontos percentuais em comparação ao ParamopamaCRF, confirmando novamente a superioridade de modelos de *Deep Learning*.

**Table 8. Resultados dos modelos testados no experimento (Test Set 1)**

<b>Modelo</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
ParamopamaCWNN	83.97%	78.39%	80.50%
ParamopamaWNN	<b>86.45%</b>	<b>89.77%</b>	<b>88.08%</b>

**Table 9. Comparação com resultados do trabalho [Júnior et al. 2015] (Test Set 1)**

<b>Modelo</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
ParamopamaCWNN	83.97%	78.39%	80.50%
ParamopamaWNN	<b>86.45%</b>	<b>89.77%</b>	<b>88.08%</b>
ParamopamaCRF	79.91%	79.66%	79.79%

**Table 10. Resultados do modelo ParamopamaCWNN (Test Set 1) por entidade nomeada**

<b>Entidade</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
Pessoa	91.01%	84.16%	87.45%
Local	84.09%	83.06%	83.57%
Organização	59.04%	66.92%	62.73%
Tempo	88.86%	59.11%	71.00%

**Table 11. Resultados do modelo ParamopamaWNN (Test Set 1) por entidade nomeada**

<b>Entidade</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
Pessoa	83.76%	90.50%	87.00%
Local	87.55%	88.09%	87.82%
Organização	69.55%	82.35%	75.41%
Tempo	86.96%	89.06%	88.00%

**Table 12. Resultados dos modelos testados no experimento (Test Set 2)**

<b>Modelo</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
ParamopamaCWNN	<b>85.25%</b>	80.71%	82.51%
ParamopamaWNN	83.83%	<b>88.65%</b>	<b>86.17%</b>

**Table 13. Comparação com resultados do trabalho [Júnior et al. 2015] (Test Set 2)**

<b>Modelo</b>	<b>Precision</b>	<b>Recall</b>	<b>F-measure</b>
ParamopamaCWNN	<b>85.25%</b>	80.71%	82.51%
ParamopamaWNN	83.83%	<b>88.65%</b>	<b>86.17%</b>
ParamopamaCRF	81.26%	81.90%	81.58%

## 5. Conclusão

Neste artigo foi mostrado que a utilização de redes neurais profundas na tarefa de classificação de entidades nomeadas é recomendável, obtendo ótimo desempenho sem a ne-

**Table 14. Resultados do modelo ParamopamaCWN (Test Set 2) por entidade nomeada**

Entidade	Precision	Recall	F-measure
Pessoa	90.56%	78.89%	84.32%
Local	76.39%	90.14%	82.70%
Organização	74.83%	68.62%	71.59%
Tempo	87.73%	67.98%	76.60%

**Table 15. Resultados do modelo ParamopamaWNN (Test Set 2) por entidade nomeada**

Entidade	Precision	Recall	F-measure
Pessoa	84.36%	88.67%	86.46%
Local	84.08%	86.85%	85.44%
Organização	81.48%	54.15%	65.06%
Tempo	98.37%	87.40%	92.56%

cessidade da definição manual de *features*. O modelo proposto utilizando redes neurais recorrentes e redes neurais convolucionais demonstraram ter obtido desempenho equivalente ao trabalho de Santos (2015), estado da arte para o reconhecimento de entidades nomeadas em português do Brasil. O modelo obteve ainda desempenho superior a modelos de classificadores tradicionais, como o ParamopamaCRF. Além disso, foi mostrada a importância de usar vetores de *word embeddings* pré-treinados sobre grandes quantidades de dados, obtendo assim uma boa extração automática de diversos *features* muito importantes para o aprendizado da rede. Ressalta-se também a geração dos *char embeddings* adicionados ao modelo que, ao serem treinados em conjunto com a rede através de uma CNN, melhoraram o resultado de algumas entidades específicas como a entidade Valor. Uma investigação futura importante é o impacto negativo observado dos *char embeddings* para algumas entidades nomeadas.

## References

- Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer.
- Borthwick, A. (1999). *A maximum entropy approach to named entity recognition*. PhD thesis, New York University.
- Chiu, J. P. and Nichols, E. (2015). Named Entity Recognition with Bidirectional LSTM-CNNs. *arXiv preprint arXiv:1511.08308*.
- Chollet, F. (2015). Keras. <https://github.com/fchollet/keras>.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.
- dos Santos, C. and Guimaraes, V. (2015). Boosting Named Entity Recognition with Neural Character Embeddings. In *Proc. of NEWS 2015 The Fifth Named Entities Workshop*, page 25.

- Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.
- Friedman, N., Geiger, D., and Goldszmidt, M. (1997). Bayesian network classifiers. *Machine learning*, 29(2-3):131–163.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feed-forward neural networks. In *Int. Conf. on Artificial Intelligence and Statistics*, pages 249–256.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Júnior, C., Santos, F., Macedo, H., Chella, M., and Givigi, R. (2014). CA<sup>2</sup>JU Ilustrado uma ferramenta assistiva para crianças com paralisia cerebral. *Escola R. de Comp. Bahia-Alagoas-Sergipe*.
- Júnior, C. M., Macedo, H., Bispo, T., Santos, F., Silva, N., and Barbosa, L. (2015). Paramopama: a Brazilian-Portuguese Corpus for Named Entity Recognition. *Encontro Nac. de Int. Artificial e Computacional*.
- Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- Mikolov, T. (2012). Statistical language models based on neural networks. *Pres. at Google, Mountain View, 2nd April*.
- Murdock, J. W. (2015). Decision Making in IBM Watson™ Question Answering.
- Nothman, J., Ringland, N., Radford, W., Murphy, T., and Curran, J. R. (2012). Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence*, 194:151–175.
- Santos, D. and Cardoso, N. (2007). Reconhecimento de entidades mencionadas em português: Documentação e actas do HAREM, a primeira avaliação conjunta na área.
- Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proc. of the seventh conf. on Natural language learning at HLT-NAACL 2003-Vol 4*, pages 142–147. ACL.
- Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proc. of the 48th annual meeting of the ACL*, pages 384–394. ACL.
- Vapnik, V. (1998). The support vector method of function estimation. In *Nonlinear Modeling*, pages 55–85. Springer.
- Zaccara, R. C. C. (2012). *Anotação e classificação automática de entidades nomeadas em notícias esportivas em Português Brasileiro*. PhD thesis, Universidade de São Paulo.