ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/322745881

A Multi-Task Neural Network for Multilingual Sentiment Classification and Language Detection on Twitter

Conference Paper · January 2018

DOI: 10.1145/3167132.3167325

CITATIONS
0

READS
985

3 authors:

Jônatas Wehrmann
Pontifícia Universidade Católica do Rio Grande do Sul
16 PUBLICATIONS **78** CITATIONS
SEE PROFILE

Willian Becker
Pontifícia Universidade Católica do Rio Grande do Sul
3 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Rodrigo C. Barros
Pontifícia Universidade Católica do Rio Grande do Sul
85 PUBLICATIONS **906** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Movie Genre Classification View project

Project Efficient Neural Networks for Text Understanding View project

All content following this page was uploaded by Jônatas Wehrmann on 18 April 2018.

The user has requested enhancement of the downloaded file.

A Multi-Task Neural Network for Multilingual Sentiment Classification and Language Detection on Twitter

Jônatas Wehrmann
Pontifícia Universidade Católica
do Rio Grande do Sul
Porto Alegre, Rio Grande do Sul
jonatas.wehrmann@acad.pucrs.br

Willian E. Becker
Pontifícia Universidade Católica
do Rio Grande do Sul
Porto Alegre, Rio Grande do Sul
willian.becker@acad.pucrs.br

Rodrigo C. Barros
Pontifícia Universidade Católica
do Rio Grande do Sul
Porto Alegre, Rio Grande do Sul
rodrigo.barros@pucrs.br

ABSTRACT

In this paper, we propose a novel approach for classifying both the sentiment and the language of tweets. Our proposed architecture comprises a convolutional neural network (ConvNet) with two distinct outputs, each of which designed to minimize the classification error of either sentiment assignment or language identification. Results show that our method outperforms both single-task and multi-task state-of-the-art approaches for classifying multilingual tweets.

CCS CONCEPTS

• **Computing methodologies → Supervised learning by classification**; **Neural networks**;

KEYWORDS

multitask classification, sentiment analysis, language detection, convolutional neural networks

ACM Reference format:
Jônatas Wehrmann, Willian E. Becker, and Rodrigo C. Barros. 2018. A Multi-Task Neural Network for Multilingual Sentiment Classification and Language Detection on Twitter. In *Proceedings of ACM SAC Conference, Pau,France, April 9-13, 2018 (SAC'18),* 8 pages.
https://doi.org/https://doi.org/10.1145/3167132.3167325

1 INTRODUCTION

Nowadays, social media is already part of people's everyday life. Twitter is one of the most popular social network platforms on the Internet, where users write micro-posts (referred as *tweets*, small texts limited to 140 characters) to share their opinions on any topic. Bearing in mind the immense amount of streaming data regarding *trending* (i.e., up-to-date and heavily-commented) discussions, this particular social network has become one of the most interesting resources for a variety of applications, ranging from presidential election data analysis [29] to football fans behavior modeling [36]. One of the computational tasks that are often used for several purposes is *tweet sentiment analysis*, whose goal is to automatically identify the polarity (i.e., positive, neutral, or negative) of the tweet via natural language processing (NLP) techniques.

In a globalized world, social media analysis should not be restricted to single-language approaches, otherwise significant information regarding a given phenomenon may be lost. Specifically for Twitter, it is estimated that half of its posts are written in languages other than English[1], reaffirming the importance of multilingual strategies for tweet classification. In a nutshell, multilingual sentiment analysis approaches are usually addressed as follows:

- translating documents from their original source language to English, and then performing sentiment analysis with English-based approaches [8];
- translating documents from English to the target language of the sentiment analysis method [19];
- making use of a lexicon with sentiment-denoting words for all considered languages [10].

Translation approaches are often the preferred strategy for addressing multilingual sentiment classification. However, even if one could have perfect translation for a particular set of documents, there is also the potential cultural distance between source and target languages, which may largely influence the final classification performance [2, 32].

Previous work [6] have shown that deep neural networks are effective tools for NLP related tasks. Machine Translation [16], Sentiment Analysis [24] and Part-of-speech tagging [38] are just a few examples of NLP tasks that have greatly benefited from the effectiveness of deep neural networks. Long short-term memory networks (LSTMs) and convolutional neural networks (ConvNets) besides being state-of-the-art approaches for computer vision tasks [12, 23, 26, 30, 31, 34], they are now standard approaches for sentiment [2, 37] and text classification [5], including language identification.

In this paper, we propose to address the problem of multilingual sentiment analysis on Twitter with a multi-task deep neural network. Our architecture makes use of a character-based ConvNet to perform both sentiment analysis and language detection at the same time. To the best of our knowledge, this is the first paper to present a multi-task character-based ConvNet architecture for sentiment analysis and language detection. Moreover, we present a visual approach to analyze the impact of each character in the classification by exploring the gradients of the generated model. This data visualization scheme allows the identification of similar sentences in the multilingual Twitter dataset. Finally, we extensively evaluate our proposed method by comparing it with state-of-art

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SAC'18, April 9-13, 2018, Pau,France
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5191-1/18/04...$15.00
https://doi.org/https://doi.org/10.1145/3167132.3167325

[1] https://semiocast.com/downloads/Semiocast_Half_of_messages_on_Twitter_are_not_in_English_20100224.pdf

baselines. Results confirm that our approach is the best choice for both sentiment and language classification on Twitter.

The remainder of this paper is organized as follows. Section 2 presents an overview of related work. Section 3 details our strategy to perform sentiment analysis and language detection in a multi-task character-based fashion. Section 4 describes the experimental setup and the dataset characteristics, as well as the baseline models. Section 5 provides the results of our empirical analysis, and Section 6 concludes the paper, pointing to future work directions.

2 RELATED WORK

Sentiment classification techniques can be roughly divided into three approaches: machine learning (ML)-based, lexicon-based, and hybrid [17]. ML-based approaches address sentiment analysis as a regular text classification problem that makes use of syntactic and/or linguistic features [18]. Whereas some approaches identify the aspects that are being discussed together with their polarity (e.g., hotel reviews) [9], others simply assign an overall polarity to the entire document (e.g., movie reviews) [22].

One key aspect regarding ML-based approaches for text processing is regarding the data representation. Traditional ML algorithms are not capable of dealing with raw text, requiring some kind of feature engineering. Most studies make use of *n*-grams to represent text [22], while word-embeddings [15] and character-based representations are becoming popular as well [2, 32, 37].

Machine translation is often applied for multilingual sentiment analysis applications [1]. Mihalcea et al. [19] make use of an English corpora to train sentence-level subjectivity classifiers in Romanian language using two approaches. First, they employ a bilingual dictionary to translate an existing English lexicon to build a target language subjectivity lexicon. In the second approach, they generate a subjectivity-annotated corpus in a target language by projecting annotations from an automatically-annotated English corpus. The authors argue that both approaches can be applied to any language, and not only Romanian.

Wehrmann et al. [32] propose an efficient deep learning approach to perform multilingual sentiment analysis of four languages. A simple convolutional network with one convolutional layer, one max-pooling overtime layer, and a Softmax layer achieves state-of-the-art results while requiring only 1225× fewer parameters than the second-best baseline. The authors argue that they were the first to investigate character-based neural networks for language-agnostic translation-free sentiment classification in multilingual scenarios.

Language identification is the task of automatically detecting the language(s) present in a document based on the content of the document. There are different approaches for feature selection for language detection. Basically, these include the presence of particular characters which act as discriminators [39] or particular *n*-grams [25, 28]. Recently, language identification of short strings became a hot topic in the research community. The size of the input text is known to play a significant role in the accuracy of automatic language identification, with accuracy decreasing on shorter input documents [3]. Hammarstrom [11] presents a method that augments a dictionary with an affix table and tests it over synthetic data derived from a parallel bible corpus. In [4], different approaches for language identification are compared. The authors propose a method that relies on a decision tree to integrate outputs from several different language-identification approaches. In [35], the authors propose a language identification system based on a Hidden Markov Model (HMM) for modelling character sequences. That method was used to automatically identify five languages in web documents: English, German, French, Spanish, and Italian.

3 METHOD

Feature space representation is a key aspect when using ML-based text classification in general. The most common word representation strategy consists in projecting tokenized words into dense vectors. In this scheme, each word is embedded into a d-dimensional vector space [20]. These so-called *word embeddings* are often designed to have semantically-similar words as neighbors in the multi-dimensional feature space. For instance, words *good* and *awesome* lie close to each other within the generated d-dimensional space. In a multilingual scenario, this property may allow semantically-similar words across different languages to be close in an embedded space. Notwithstanding, word-level approaches require the use of a vocabulary that stores the known words, posing two major shortcomings: (i) for multilingual tasks, the vocabulary size grows with the number of languages that are employed, which may demand large amounts of memory; and (ii) sensitivity to rare words and typos, since such words would rarely be present in the stored vocabulary.

A recent approach for learning textual representation is based on the atomic part of the words: the characters [37]. In this strategy, all characters $c_i \in \{c_1, c_2, ..., c_t\}$ in $\mathcal{T}$ are quantized into a binary matrix of size $t \times \eta$, for an alphabet with η characters. Recently, researchers have experimented the use of dense vectors for a more compact representation of the characters [7, 33].

The main advantage of the character-based approach is that the cost of text representation depends only on the number of characters in the text t and the number of known characters η. For instance, a language-agnostic classifier that learns across four syntactically similar languages (e.g, Portuguese, Spanish, English, and German) requires only ≈ 80 different characters for representing the whole content of the languages. On the other hand, by using a word-embedding-based method the minimum cost for the data representation alone is given by $w \times (\mathcal{V}_{English} + \mathcal{V}_{German} + \mathcal{V}_{Portuguese} + \mathcal{V}_{Spanish})$. Assuming that $|\mathcal{V}|$ for all languages is similar, and contains $\approx 50,000$ words, the inherent cost will be $t \times 200,000$. Even though t is often 5× larger than w, a character-level representation presents an advantage of about $\frac{t \times 200,000}{5 \times t \times 80} = 500\times$.

In this paper we introduce NNLS, which is an architecture that is capable of learning two tasks at once: multilingual sentiment analysis and language identification. Our approach relies solely on the use of character-level convolutions. This combination provides more robustness to noisy data, allowing to save memory since it does not process a pre-defined vocabulary, and it is faster given that convolving temporal data is easily parallelized. For instance, the i^{th} temporal iteration of a recurrent neural network depends on the $(i-1)^{th}$ iteration, while the temporal information learned by convolutions are fused along the network's depth.

We introduce two variations of NNLS, namely NNLS-v1 and NNLS-v2. They consist of three flows of at least one convolutional layer that processes the text input by using different filter sizes. Hence, one can learn information from character-level input in an explicit hierarchical manner. This architectural choice makes it easier for the network to learn both fine-grained and mid-level information. Hence, it is possible to encode most of the text semantics related to sentiment analysis, while providing important information regarding the document language. We do believe that by forcing the network to detect the language as well, it can then more easily learn specific features regarding each language's sentiment information. In addition, trying to approximate a shared function for distinct tasks naturally constrain the learning, given that the learned weights must work for both tasks, possibly introducing a regularizing effect.

Figure 1 shows the overall architecture of NNLS-v1. In this version we use only three parallel convolutional layers that convolve the input by applying three different filter sizes (similarly to [15, 26]). This enables the network to learn up to trigram-level features, depending on the word length.

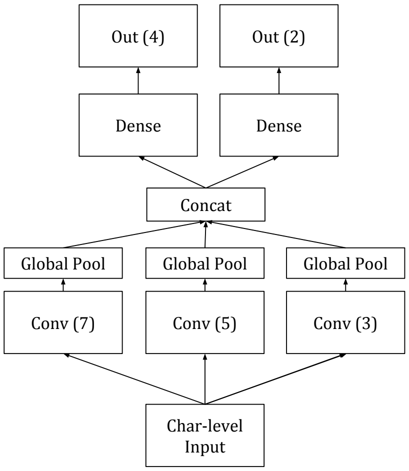

Figura 1: Overall structure of NNLS-v1.

NNLS-v2 is illustrated in Figure 2. It is quite similar to the first version, but following [27] the 7 × η convolutions are replaced by three 3 × η convolutions, and 5 × η layers are replaced with two 3 × η convolutions. This replacement provides the same receptive field while increasing non-linearity and reducing the number of parameters.

Overall, the first convolutional stack in both NNLS-[v1, v2] processes the input text by applying three convolutional layers that help increasing the receptive field and the non-linearity, being the responsible for learning mid-level information. This can be seen as a sophisticated way to learn n-grams. The second convolutional stack is designed to learn short-term information, similarly to bigrams (depending, of course, on the word length). The last one processes the input by applying a single convolutional layer, which can learn fine-grained information, making it easier to leverage the existence of short character sequences (e.g., *:), :/, :-D, omg, wtf, 4you, gooood*, etc).

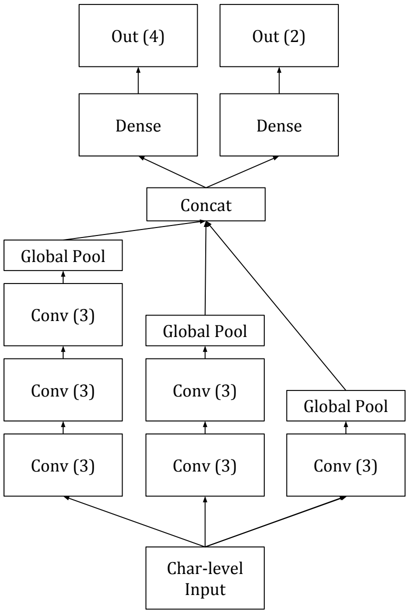

Figura 2: Overall structure of NNLS-v2.

All convolutional stacks are followed by a global max-pooling, which selects the most important features across the temporal dimension, increasing the receptive field to the entire input. This pooling strategy results in vectors $\mathbf{f} \in \mathbb{R}^k$, where k is the number of filters of the respective stack. Hence, our architecture generates 3 independent $\mathbf{f}$ vectors. The final shared textual representation $\mathcal{F}$ is built by concatenating all 3 $\mathbf{f}$ vectors, explicitly encoding fine-grained, short-term, mid-level, and global features in a hierarchical fashion.

The final part of the architecture consists of two two-layer neural networks that are not shared across tasks, although both are fed with $\mathcal{F}$. This helps to increase the non-linearity needed for learning particular features for each task. Hidden layers from all networks comprise ReLU neurons. The final layer is responsible for linearly mapping the hidden representation to the specific classes of both tasks. Hence, the first network output generates four classes for identifying each language we are learning from (considering a 4-language detection problem), and the second one learns binary sentiment analysis (positive or negative sentiment). Note that our overall architecture ends with two softmax outputs, normalizing the scores into probability distributions.

3.1 Loss Function

Let $\phi_L(\mathcal{F}) = \mathbf{v}$ be the function that approximates the language identification by generating $\mathbf{v}$ scores, and $\phi_S(\mathcal{F}) = \mathbf{z}$ be the scores for the sentiment analysis task. Both functions are based on a global shared function learned directly form the character-level text input $\mathcal{T}$ through the convolutional layers, namely $\phi_G(\mathcal{T}) = \mathcal{F}$.

Tabela 1: Tweet corpora. Each of the sets contain positive and negative tweets from 4 languages.

	Training		Validation		Test		
Language	Negative	Positive	Negative	Positive	Negative	Positive	Total
English	7,764	7,765	1,122	1,085	2,229	2,205	22,170
German	7,476	10,570	1,062	1,549	2,075	3,089	25,821
Portuguese	17,248	12,228	2,484	1,715	4,854	3,565	42,094
Spanish	8,289	18,380	1,158	2,643	2,324	5,294	38,088
Multilingual	40,777	48,943	5,826	6,992	11,482	14,153	128,173
Amount of the total corpora	89,720 (70%)		12,818 (10%)		25,635 (20%)		

Tabela 2: Examples of tweets from the multilingual dataset.

Language	Text	Class
English	*Have this abstract Om Painting to give beautiful vibrant colours to wall. http://t.co/ZSoQOYN8xD*	Positive
English	*Ok i dont want to wake up*	Negative
German	*Das schöne an Osnabrück? Kostenfreies WLAN in der ganzen Innenstadt!*	Positive
German	*Da haben die Eltern echt mächtig was falsch gemacht! :O*	Negative
Portuguese	*lol, é smp assim. em vez de tar a estudar tou no twitter :)*	Positive
Portuguese	*To com moh vontade de te exclui sabiia?!*	Negative
Spanish	*Me gustan los abrazos sin motivo.*	Positive
Spanish	*Me duele la cabeza.*	Negative

To approximate those functions, we need to maximize the log-likelihood of the generated scores $\mathbf{s}_T = \{\mathbf{v}, \mathbf{z}\}$ of the correct class y given $\mathcal{T}$, by

$$P_T(y = j|\mathcal{T}) = \frac{e^{\mathbf{s}_{T_j}}}{\sum_{k=1}^{K} e^{\mathbf{s}_{T_k}}} \tag{1}$$

Given the $P_T(y = j|\mathcal{T})$ probabilities for $T \in \{L, S\}$ tasks, we minimize the negative log-likelihood for both T tasks:

$$\mathcal{L}_T = -\log(P_T(y = j|\mathcal{T})) \tag{2}$$

where $\mathcal{L}_L$ and $\mathcal{L}_S$ denote, respectively, the the loss function for the language identification and sentiment analysis tasks.

Finally, the overall loss function $\mathcal{L}_G$ is given by

$$\mathcal{L}_G = \alpha\mathcal{L}_L + \beta\mathcal{L}_S \tag{3}$$

where α and β regulate the trade-off between both tasks. Here, larger α values cause the language identification task to have greater importance in the optimization process, while larger β values make the network more sensitive to the performance in the sentiment analysis task. Refer to Section 5.1 for an empirical analysis with a detailed discussion regarding the impact of α and β.

4 EXPERIMENTAL SETUP

In this section we detail methodological aspects of the experimental analysis for allowing reproducibility, such as the dataset we employ, the baseline approaches, hyper-parameter settings, and training protocol.

4.1 Dataset

We make use of the Twitter corpora from [21] to evaluate the proposed architecture. This dataset contains around 1.6 million annotated tweets from 13 European languages, and it is considered one of the largest corpora publicly available nowadays. All tweets have been manually labeled into three classes: *positive*, *neutral*, and *negative*. Due to the semantic and syntactic structural differences among the 13 languages, we only consider a subset of tweets from four specific languages: English, Spanish, Portuguese, and German. See Table 1 for statistics of the dataset, and Table 2 for a few samples extracted from it. We also reduce the problem to binary classification by discarding all neutral tweets. Note that the dataset does not provide the tweet itself, but rather a URL that leads to the tweets. Due to this particularity, some tweets are no longer available. To build a language detection dataset, we manually labeled each instance of the dataset according to its corresponding language.

4.2 Baseline Algorithms

We compare our results with the following algorithms: (i) LSTMs [13], the standard deep neural network approach for handling temporal data; (ii) ConvChar [37], the first work to learn textual representations from raw characters; (iii) ConvEmb [15], a fast architecture that applies a convolutional layer with multiple filter sizes, followed by a global max-pooling; (iv) FastText [14], an efficient convolutional network for text classification; and (v) ConvChar-S [32], a neural network that convolves raw characters and achieves state-of-the-art results for multilingual sentiment analysis.

For the sake of fairness, we do not apply any preprocessing to the data. All models were trained with the original raw data. For the word-embedding-based approaches, the vocabularies are built considering all words present in the training set.

Several authors [2, 32] have shown that traditional hand-crafted features do not work well for multilingual sentiment analysis. Hence, we will not use them for comparison. In addition, since the dataset is reasonably balanced among the classes, results are reported in terms of classification accuracy.

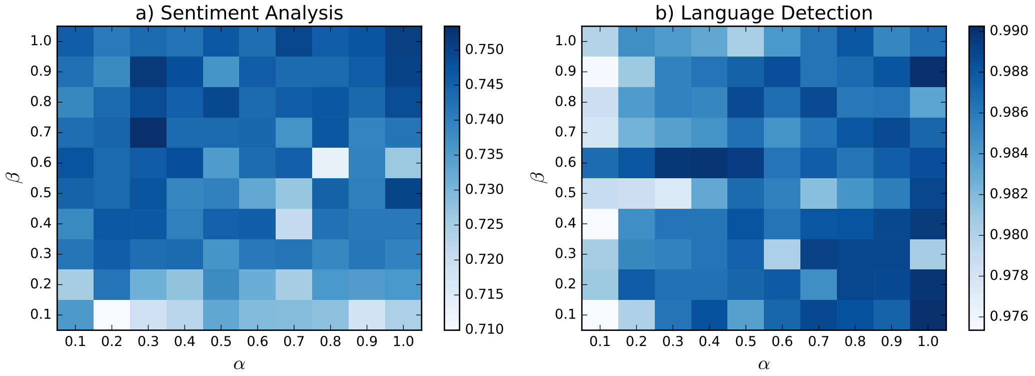

Figura 3: Loss weights

4.3 Hyper-Parameters

The hyper-parameters of the baseline approaches are set following their original implementations. For the cases where such values are unknown, we follow those used in [32]. Namely, for word-embedding-based networks, we use $d = 300$. Character-level networks are trained with $\eta = 70$ characters. Since we are classifying *tweets*, the number of characters is fixed in 140, while the word-based methods process fixed-length texts where the maximum number of words is defined by the sentence in the training set with the largest number of words.

- LSTM [13]: single hidden layer with 512 units.
- ConvChar [37]: we use the very same architecture from the original implementation, though we achieve better results training with the Adam optimizer, weight initialization as in [12], and learning rate of 1×10^{-3}. We also use a larger dropout rate, as in [32], in order to better regularize the network.
- ConvEmb [15]: we use same settings of the original work.
- FastText [14]: we use same hyper-parameters of the original work, though we employ only unigram features for not exploding the memory requirements.
- ConvChar-S [32]: we use its original implementation for replicating the results.
- NNLS-[v1, v2]: we use both 128 and 200 (default) convolutional filters per layer, referred as NNLS-vk(number of filters).

All models are trained for a maximum of 100 epochs, early-stopping when validation accuracy stop improving for 5 consecutive epochs. All models are trained with a learning rate of 1×10^{-3} and optimized using the Adam update rule, which performs per-weight learning rate annealing, and hence we do not perform any additional learning rate reduction.

Note that all multi-task models are trained with the loss function defined in 3.1. For finding the best values of both α and β, we performed a grid-search $\in \{0.01, 0.02, 0.03, ..., 1.00\}$. Such a procedure required us to train at least 100 models per approach, resulting in a total of 500 fully-trained networks. This took about a week running experiments in a single server equipped with 4 GPU Tesla 1080Ti 11GB (Pascal architecture). The loss hyper-parameters were chosen based on validation accuracy.

5 EXPERIMENTAL ANALYSIS

In this section we provide an extensive set of experiments for evaluating some architectural choices and hyper-parameters. We first analyze the impact of the weights in our multi-task loss function. Next, we compare our architecture with the state-of-the-art approaches. In addition, we show qualitative examples of tweets retrieved using features embedded in different levels of the network, and finally, we use gradients of the network for visualizing and understanding our models.

5.1 Loss weights

In this section we analyze the impact of the loss weights in our architecture, namely α and β.

Figure 3 depicts the results of the 100 models trained in the grid-searching procedure. As expected, smaller values of α imply in worse language detection performance, and smaller values of β tend to generate poor results for sentiment analysis. However, one can observe that some loss weights settings help in achieving good results for both tasks. By verifying the predictive performance in the validation set for both tasks, we chose values of $\alpha = 0.9$ and $\beta = 1.0$. Despite the fact that this configuration does not generate the best global results for both tasks, it is the setting that maximizes the conjoint performance.

5.2 Comparison with the state-of-the-art

Since we designed a multi-task neural network architecture that learns conjointly two distinct tasks, we provide results for two training strategies: (i) we train all networks separately for each task to evaluate their full capacity for learning those tasks; and (ii) we train all baselines approaches adapting them to be multi-task, similarly to our approach.

We perform several experiments to evaluate our models and compare them with the baselines. Table 3 presents the results for all methods when adapting the baselines to perform multi-task learning.

Tabela 3: Multi-task learning results for all methods.

Method	Sentiment Analysis	Language Detection
LSTM	71.33%	97.62%
ConvEmb	71.75%	97.74%
FastText	71.31%	97.52%
ConvChar	70.59%	97.55%
ConvChar-S	73.37%	98.37%
NNLS-v1	**73.63%**	98.11%
NNLS-v2	**74.43%**	**98.40%**

Our models outperform all baselines in the multi-task scenario. The baseline character-based models also reach good results in both tasks. ConvChar comprises several convolutions through its structure increasing the model non-linearity, which could be especially interesting when working in a multilingual scenario. However, our proposed model has fewer convolutional and pooling layers, and improves classification accuracy in relative terms by ≈ 5.15% for sentiment analysis and ≈ 0.87% for language detection. Results of the ConvChar-S architecture show that this model is capable of performing both tasks at the same time with quite competitive results when compared with our approach. Overall, our models have proven to be superior, as illustrated in Figure 4, which presents all trained models and their respective performance in both tasks.

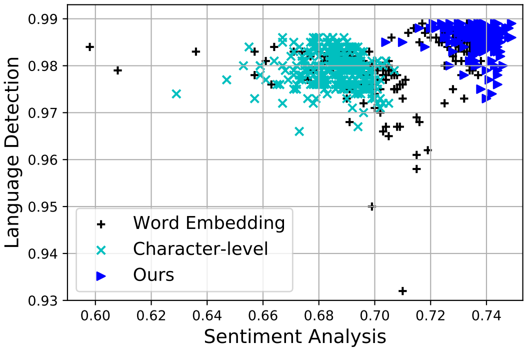

Figura 4: All trained models.

When comparing our model with word-embeddings-based approaches, we note that the time-consuming phase of learning textual representation does not reflect in better results. LSTM, FastText, and ConvEmb reach similar results in both tasks, with LSTM providing slightly better accuracy than the other embedding-based models.

5.2.1 Single-Task Learning. All baseline models where also evaluated only for language detection, or only for sentiment analysis. These experiments aim to analyze whether the baseline methods can outperform our approaches when considering single-task learning.

Table 4 presents results regarding language detection and sentiment analysis as single-tasks. In this scenario, we trained each classifier to perform only one of the tasks, not sharing network weights across the tasks.

Tabela 4: Single-task learning results for all methods.

Method	Sentiment Analysis	Language Detection
LSTM	71.57%	**98.43%**
ConvEmb	71.19%	98.38%
FastText	71.39%	98.01%
ConvChar	70.71%	96.91%
ConvChar-S	71.78%	97.61%
NNLS-v1(128)	**72.52%**	97.72%
NNLS-v1(200)	**72.87%**	97.53%
NNLS-v2(128)	**72.97%**	97.59%
NNLS-v2(200)	**73.55%**	97.83%

Our model outperforms all baselines in single-task sentiment analysis, whereas LSTMs reaches the best accuracy for language detection when training for it as a single-task.

5.2.2 Per-Language Sentiment Analysis. We also investigate the performance of all methods in per-language sentiment analysis, which means we train all models in a multilingual fashion but we then test their performance in per-language datasets. Table 5 presents the results of this analysis.

Tabela 5: Per language results of SA by training with multilingual data and testing with per language datasets.

Method	English	German	Portuguese	Spanish
LSTM	70.57%	76.58%	71.12%	69.32%
ConvEmb	72.30%	**77.58%**	69.28%	69.50%
FastText	70.04%	77.22%	71.01%	69.15%
ConvChar	70.24%	72.07%	71.24%	69.74%
ConvChar-S	74.97%	75.50%	74.36%	69.83%
NNLS-v1	**75.35%**	76.31%	**74.86%**	**70.01%**
NNLS-v2	**76.68%**	76.72%	**75.13%**	**70.02%**

5.3 Network Visualization

In this section we use the method proposed in [32] for visualizing features learned by the network. This visualization technique consists of back-propagating gradients to the character-level input. Such information is normalized and plotted to highlight the character sequences that were most important for providing a given sentiment prediction. All visualizations in this section are resulting of our best architecture, namely NNLS-2. Figures 5 and 6 show examples of correctly classified tweets from the positive and negative polarities (validation data). Character sequences in red depict the most relevant parts of the tweet for the corresponding prediction. Note that our method can easily leverage the existence of emoticons, some particular words, and sequences of words as well.

5.4 Multi-Task Embedding Space

A final qualitative analysis concerns retrieving the most similar tweets by using latent features optimized by training two tasks at once. More specifically, given a query tweet we extract the $\mathcal{F}$ features and compare to the $\mathcal{F}$ features extracted from 1,000

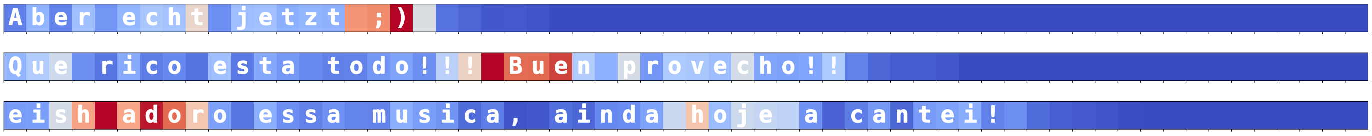

Figura 5: Positive sentences from the Twitter multilingual dataset.

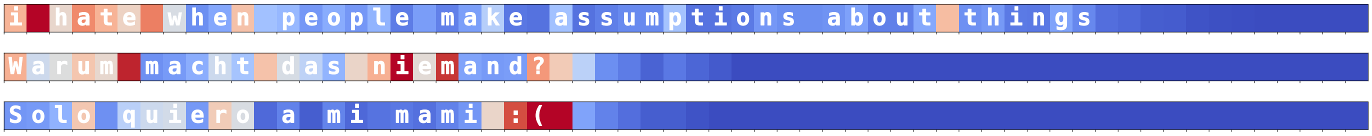

Figura 6: Negative sentences from the Twitter multilingual dataset.

Tabela 6: Analysis of the multi-task embedding space. Retrieved tweets from query sentences.

Query	Retrieved Sentences
Alot of enjoyment on this weekend. ! :-D	RT @usr: photography is such a great thing in life :-) @usr i got the same one :D @usr Love U!!! Please tweet me, i just wanna that happen :D
não consigo mais, acho que já morri de tanto estudar	Não peço muito, só ser rica e não ter que ir à escola Não oiço certas musicas porque me lembram certas coisas não vou deixar que o pior aconteça só porque estás a ser parvo.
Ah, das neue Mo Hayder Buch ist da.	@usr Sehr gut! XD @usr ja. Sachsenhausen ist gut! Du suchst sicher was schönes raus :-) @usr @usr ok, wenn's ein trinkspiel gibt bin ich *definitiv* dabei!
@usr Muchas gracias Paloma :)	@usr buenos días :D @modery Gratuliere ;) @Ivihermanostodo Hoy comemos juntos ;)

randomly selected tweets from the validation data. All features are normalized to have unit norm so that the vector multiplication results in the cosine distance. Hence, we use the cosine for ranking the most similar tweets. Table 6 depicts examples of retrieved tweets given four queries, one query per language. Note that by using the features optimized via our multi-task loss function, we are able to automatically retrieve tweets from the same language and the same polarity. Moreover, some retrieved tweets are quite similar in structure when compared to the query sentence.

6 CONCLUSION

In this paper we proposed a multi-task neural network architecture for performing sentiment analysis and language detection at the same time. Our architecture outperforms all state-of-the-art approaches by a good margin. In addition, our networks are light and fast, requiring about two orders of magnitude fewer parameters than the word-embedding baselines. For future work, we intend to compare our approach with translation approaches, as well as increasing the number of languages and including distinct alphabets (e.g., Chinese or Japanese).

ACKNOWLEDGMENTS

We would like to thank Google, Motorola and the Brazilian research agencies CAPES, CNPq, and FAPERGS for funding this research. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

REFERÊNCIAS

[1] Carmen Banea, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. 2008. Multilingual subjectivity analysis using machine translation. In *EMNLP*. 127–135.
[2] Willian Becker, Jonatas Wehrmann, Henry EL Cagnini, and Rodrigo C Barros. 2017. An Efficient Deep Neural Architecture for Multilingual Sentiment Analysis in Twitter. In *FLAIRS*. 246–251.
[3] William B Cavnar, John M Trenkle, et al. 1994. N-gram-based text categorization. *Ann Arbor MI* 48113, 2 (1994), 161–175.
[4] Hakan Ceylan and Yookyung Kim. 2009. Language identification of search engine queries. In *ACL*. 1066–1074.
[5] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*. 160–167.
[6] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, Aug (2011), 2493–2537.
[7] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781* (2016).
[8] Kerstin Denecke. 2008. Using sentiwordnet for multilingual sentiment analysis. In *International Conference on Data Engineering*. 507–512.
[9] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric Ringger. 2005. Pulse: Mining customer opinions from free text. In *IDA*. 121–132.
[10] Alexandru-Lucian Gînscă, Emanuela Boroş, Adrian Iftene, Diana TrandabĂţ, Mihai Toader, Marius Corîci, Cenel-Augusto Perez, and Dan Cristea. 2011. Sentimatrix: multilingual sentiment analysis service. In *Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*. 189–195.
[11] Harald Hammarström. 2007. A fine-grained model for language identification. In *Proceedings of iNEWS-07 Workshop at SIGIR 2007*. 14–20.
[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *ICCV*.
[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).

[15] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[16] Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *arXiv preprint arXiv:1610.03017* (2016).

[17] Diana Maynard and Adam Funk. 2011. Automatic detection of political opinions in tweets. In *Extended Semantic Web Conference*. 88–99.

[18] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal* 5, 4 (2014), 1093–1113.

[19] Rada Mihalcea, Carmen Banea, and Janyce M Wiebe. 2007. Learning multilingual subjective language via cross-lingual projections. (2007).

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*.

[21] Igor Mozetič, Miha Grčar, and Jasmina Smailović. 2016. Multilingual Twitter Sentiment Classification: The Role of Human Annotators. *PloS one* 11, 5 (2016), e0155036.

[22] Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*. 271.

[23] Gabriel S Simões, Jônatas Wehrmann, Rodrigo C Barros, and Duncan D Ruiz. 2016. Movie genre classification with convolutional neural networks. In *IJCNN*. 259–266.

[24] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*. 1631–1642.

[25] Clive Souter et al. 2017. Natural language identification using corpus-based models. *HERMES-Journal of Language and Communication in Business* 7, 13 (2017), 183–203.

[26] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *CVPR*.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the Inception Architecture for Computer Vision. *CoRR* abs/1512.00567 (2015).

[28] Tommi Vatanen, Jaakko J Väyrynen, and Sami Virpioja. 2010. Language Identification of Short Text Segments with N-gram Models.. In *LREC*.

[29] Hao Wang, Dogan Can, Abe Kazemzadeh, François Bar, and Shrikanth Narayanan. 2012. A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle. In *Proceedings of the ACL 2012 System Demonstrations (ACL '12)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 115–120.

[30] Jônatas Wehrmann and Rodrigo C Barros. 2017. Movie Genre Classification: A Multi-Label Approach based on Convolutions through Time. *Applied Soft Computing* in press (2017).

[31] Jônatas Wehrmann, Rodrigo C. Barros, Gabriel Simões, Thomas S. Paula, and Duncan D. Ruiz. 2016. (Deep) Learning from Frames. In *BRACIS*. 6.

[32] Jonatas Wehrmann, Willian Becker, Henry EL Cagnini, and Rodrigo C Barros. 2017. A character-based convolutional neural network for language-agnostic Twitter sentiment analysis. In *IJCNN*. IEEE, 2384–2391.

[33] Jônatas Wehrmann, Anderson Mattjie, and Rodrigo C Barros. 2017. Order embeddings and character-level convolutions for multimodal alignment. *arXiv preprint arXiv:1706.00999* (2017).

[34] Jônatas Wehrmann, Gabriel S. Simões, Rodrigo C. Barros, and Victor F. Cavalcante. 2017. Adult content detection in videos with convolutional and recurrent neural networks. *Neurocomputing* in press (2017).

[35] Alexandros Xafopoulos, Constantine Kotropoulos, George Almpanidis, and Ioannis Pitas. 2004. Language identification in web documents using discrete HMMs. *Pattern recognition* 37, 3 (2004).

[36] Yang Yu and Xiao Wang. 2015. World Cup 2014 in the Twitter World: A big data analysis of sentiments in U.S. sports fans' tweets. *Computers in Human Behavior* 48, Supplement C (2015), 392 – 400.

[37] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

[38] Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep Learning for Chinese Word Segmentation and POS Tagging.. In *EMNLP*. 647–657.

[39] D.V. Ziegler. 1991. *The Automatic Identification of Languages Using Linguistic Recognition Signals*. State University of New York at Buffalo.

View publication stats