Text Language Identification Using Attention-Based Recurrent Neural Networks

Conference Paper · May 2019

DOI: 10.1007/978-3-030-20912-4_18

2 authors:

Michał Perełkiewicz
Ośrodek Przetwarzania Informacji
8 PUBLICATIONS 0 CITATIONS
SEE PROFILE

Rafał Poświata
Ośrodek Przetwarzania Informacji
6 PUBLICATIONS 1 CITATION
SEE PROFILE

Text Language Identification Using Attention-Based Recurrent Neural Networks

Michał Perełkiewicz, Rafał Poświata

National Information Processing Institute, Warsaw, Poland
{mperelkiewicz, rposwiata}@opi.org.pl

Abstract. The main purpose of this work is to explore the use of Attention-based Recurrent Neural Networks for text language identification. The most common, statistical language identification approaches are effective but need a long text to perform well. To address this problem, we propose the neural model based on the Long Short-Term Memory Neural Network augmented with the Attention Mechanism. The evaluation of the proposed method incorporates tests on texts written in disparate styles and tests on the Twitter posts corpus which comprises short and noisy texts. As a baseline, we apply a widely used statistical method based on a frequency of occurrences of n-grams. Additionally, we investigate the impact of an Attention Mechanism in the proposed method by comparing the results with the outcome of the model without an Attention Mechanism. As a result, the proposed model outperforms the baseline and achieves 97,98% accuracy on the test corpus covering 36 languages and keeps the accuracy also for the Twitter corpus achieving 91,6% accuracy.

1 Introduction

A text language identification problem (language ID) refers to the process of determining language based on a text structure under a given classification system. Language ID is often found as the first step in commonly used applications like text translators, web search engines, or Twitter (used in data stream tagging with appropriate language), affecting further outcomes significantly.

The main purpose of this study is to investigate the predictive performance of an Attention-based Recurrent Neural Network (ARNN) in comparison with a standard Recurrent Neural Network (RNN) and a statistical, n-gram-based approach, in a language identification problem. The use of a Bidirectional Long Short-Term Memory Neural Network (BiLSTM) model augmented with an Attention Mechanism is motivated by employing this approach successfully in other NLP tasks, such as text classification [5], language translation [12], and relation classification [15]. LSTM neural networks are capable of handling long-term dependencies in a sequential type of data. Such a type of neural networks is designed to avoid long-term dependency problems, like vanishing gradient [7]. Thus, this model is suitable for working with sequential text data to extract relations occurred in data.

The most problematic issues for text language identification systems are working with short texts, texts with unconventional spelling and written in an informal style, with grammatical and syntax errors and closely related language pairs. We show that the proposed method performs well for this kind of troublesome texts, even with learning only on readily available, well-formatted corpora.

The remainder of the paper is structured as follows. In Section 2, we review related work about language identification. Section 3 presents the proposed Attention-based BiLSTM model in detail. In section 4, we describe datasets used to train and validate the model and present the results of the proposed method and two baseline methods. Finally, conclusions are included in Section 5.

2 Previous work

In the past, many different methods have been used to address the problem of text language identification. Cavnar and Trenkle [2] employed a statistical, character-based n-gram model built upon the most frequent 1 to 5-grams in a text. Variants on this approach incorporate Bayesian models (for example CLD2[1] - the language identification tool created by Google in 2013), dot products of word frequency vectors [4], different measures of document similarity and the distance between n-gram profiles [1, 14]. Other statistical approaches applied in language identification base on Markov models [11], kernels methods in SVMs [10]. Grefenstette [6] used a word and a part of speech (POS) correlation to determine if two text samples were written in the same or different languages. These methods are widely used in many NLP programming libraries, like Cybozu Labs Language-Detection library[2], Optimize Language Detection [3] for Java and langid.py[4] for Python. The main drawback of these statistical methods is a low efficiency in working with short texts. Other disadvantages of these language identification approaches are a high impact of foreign words occurred in the analysed text and tendency to predict errors when working with noisy text. Statistical methods based on n-grams ignore long-term relationships between characters occurring in a text.

One of the first uses of Neural Networks to address the language identification problem was presented by Chang and Lin [3]. This approach, employing a RNN and skipgram word embedding, outperformed the top results for English-Spanish and English-Nepal language pairs identification competition in the EMNLP 2014 Language Identification in Code-Switched Data[5]. Other approach based on a neural architecture [8] exploits the model build of two main components. The first is a Convolutional Neural Network (CNN) to delimit a whitespace word's Unicode character sequence. The second is a BiLSTM recurrent neural network that maps a sequence of word vectors to a language label. Kocmi and Bojar [9]

[1] https://github.com/CLD2Owners/cld2
[2] https://github.com/chbrown/language-detection
[3] https://github.com/optimaize/language-detector
[4] https://github.com/saffsd/langid.py
[5] http://emnlp2014.org/

used BiLSTM Recurrent Neural Network to operate on a window of 200 Unicode characters of the input text. This model outperformed statistical models for language identification based on a short text. These neural approaches can be predictive mistake-prone for unseen text structures or text containing foreign words. Adding an Attention Mechanism to the RNN model is a potential solution to these problems.

3 Proposed method

We adapt the BiLSTM recurrent neural architecture proposed by [9] with some changes. Firstly, the model we propose is built upon the 2-layer BiLSTM Neural Network instead of the 1-layer Bidirectional Gated Recurrent Units (GRUs) model. Secondly, we add the soft Attention Mechanism on top of the BiLSTM layers.

As input, the model takes a vector of a Unicode's character sequence[6]. We use one-hot embedding for character sequences, so if C is the set of unique characters in a dataset, then we let the size of the character embedding be $d = |C|$. For given input sequence, the embedded input vector A is defined as $A = [x_0, x_1, ..., x_T]$, where T is the sentence length. Each vector element x_t represents one Unicode character in the input sentence. The responsibility of the 2-layer BiLSTM Neural Network is to learn a sequential relationship between characters for each of given languages by optimising the weights in a hidden layer h_t at time step t. The hidden layer h_t is calculated based on the current input layer x_t and the previous state of the hidden layer h_{t-1}, according to the definition:

$$h_t = tanh(Wx_t + Vh_{t-1} + b_1), \tag{1}$$

where W, V are the weight vectors connected to the input vector and the previous hidden state vector respectively, and b_1 is the bias vector. The output is calculated according to the formula:

$$y_t = f(Uh_t + b_2), \tag{2}$$

where U is the matrix of weights connected to the hidden state vector, b_2 is the bias vector, and f is an activation function.

In case of a bidirectional LSTMs network (BiLSTM), not only the previous hidden state h_{t-1} is taken into account during calculating the hidden state h_t, but also the next hidden state h_{t+1} which is calculated by reading the input also from the end by the bidirectional layer. The BiLSTM model contains two hidden states, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, for each neural cell. Therefore, we extend previous calculations as follows:

$$\overrightarrow{h_t} = tanh(\overrightarrow{W}x_t + \overrightarrow{V}h_{t-1} + \overrightarrow{b_1}) \tag{3}$$

$$\overleftarrow{h_t} = tanh(\overleftarrow{W}x_t + \overleftarrow{V}h_{t+1} + \overleftarrow{b_1}) \tag{4}$$

$$y_t = f(\overrightarrow{U}\overrightarrow{h_t} + \overleftarrow{U}\overleftarrow{h_t} + b_2), \tag{5}$$

[6] The input vector we use contains 100 characters.

where the left and the right arrows indicate the reading direction of the input vector A. The output is calculated on the basis of the weighted sum of the $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ hidden states and $\overrightarrow{U}$ is the matrix of weights connected to the hidden state vector $\overrightarrow{h_t}$ and $\overleftarrow{U}$ is the matrix of weights connected to the hidden state vector $\overleftarrow{h_t}$.

After the deep BiLSTM neural model processing, the attention mechanism is responsible for deciding which characters and relations in the given sentence context indicate the output language to a greater or lesser extent.

Let H be a matrix consisting of output vectors $Y = [y_1, y_2, ..., y_T]$ that the BiLSTM layer produced, where T is the sentence length. For given vector Y, an attention-based model computes a *context* vector c_t as the weighted mean of the state sequence Y as follows [13]:

$$c_t = \sum_{j=1}^{T} \alpha_{tj} y_j \tag{6}$$

where α_{tj} is a weight computed at each time step t for each state y_j. Then, the context vectors are used to compute a new state sequence s, where s_t depends on s_{t-1}, c_t and the model's output at $t-1$. The weightings α_{tj} are then computed as follows [13]:

$$e_{tj} = a(s_{t-1}, y_j), \alpha_{tj} = exp(e_{tj}) \sum_{k=1}^{k=T} exp(e_{tk}), \tag{7}$$

where a is a learned function, which can be thought of as computing a scalar importance value for given y_j and the previous state s_{t-1}.

As the output, we use a dense neural layer with the softmax activation function. The output is the vector of probabilities over all languages classes. Let $R = [r_0, r_1, ..., r_t]$ be the output vector, where $T = |R|$ and $\sum_{i=0}^{i=T} r_i = 1$. As a result, we choose the language with the highest probability. The model is depicted in the Figure 1.

4 Experimental studies

4.1 Datasets

Text is characterised by many features, including the formality of used vocabulary, grammatical and stylistic correctness, grammatical structures, length, and so on. Building a suitable, multilingual text corpus which covers many text types is the crucial step in a learning and evaluating process. For this purpose, we focused on finding multilingual and diverse text corpora. The data set we used to learn and evaluate the model comprises 6 text corpora:

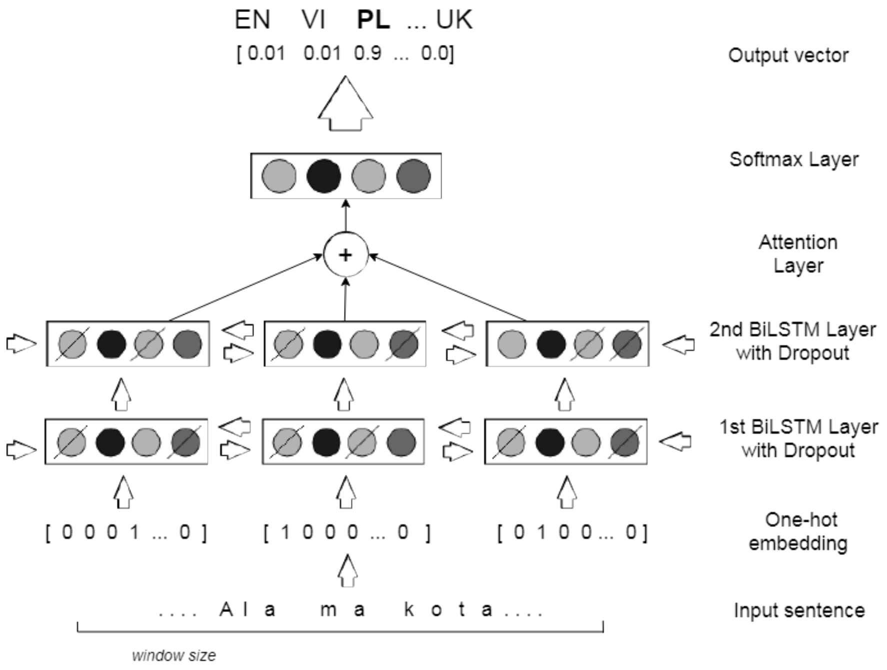

Fig. 1. The illustration of the proposed model

1. Subtitles — the collection of translated movie subtitles,
2. Wikipedia — the collection of articles from Wikipedia,
3. Web pages — the set of sentences crawled from randomly selected web pages,
4. News — the set of sentences crawled from news websites, like bbc.com, cnn.com.
5. EuroParl -– the parallel corpus extracted from the European Parliament website,
6. Tatoeba – the set of simple sentences, created by foreign language learners.

All described corpora were mixed, keeping only languages they have in common and for which at least 400,000 sentences have been collected, taking a maximum of 100,000 sentences from each corpus. It results in the data set composed of text written in 36 languages as follows: Tatar, Maltese, Norwegian, Marathi, Hindi, Vietnamese, Croatian, Hindi, Icelandic, Czech, Arabic, Latvian, Esperanto, Macedonian, Slovenian, Ukrainian, Estonian, Slovak, Romanian, Lithuanian, Turkish, Bulgarian, Modern Greek, Swedish, Danish, Russian, Dutch, Finnish, Polish, German, English, Spanish, French, Hungarian, Italian and Portuguese. All remaining languages were rejected because of too small number of sentences in their corpus. For the final dataset, we randomly chose 200,000 sentences for each language mentioned above what results in the text corpus comprising 7,200,000 sentences in 36 languages. For testing and validation purposes 30% of data were selected and remaining 70% of the corpus served as the training set.

Additionally, we consider the Twitter dataset for testing. The Twitter test set contains 46,715 randomly selected posts and covers 14 of 36 languages chosen to learn our model. The exact structure of this set is presented in Table 1.

Table 1. The structure of the Twitter test set

language	count	language	count	language	count
Modern Greek	48	Swedish	73	Polish	129
Dutch	232	German	241	Italian	426
Turkish	873	French	1221	Russian	1245
Arabic	2790	Portuguese	3643	Hindi	3876
Spanish	7665	English	24253		

The text corpus was preprocessed before learning and validating. The preprocessing process included: removing punctuation, removing non Unicode characters, merging many whitespaces as one, converting text to lowercase.

4.2 Experimental setup

The model was trained using early stopping based on the validation set. To reduce the learning time and avoid settling the learning algorithm on an error minimum, we use ADAM optimization algorithm with batch size equal to 64. As a loss function, we use cross-entropy. The best accuracy result (97,99%) for the validation set was attained after the eleventh learning epoch. The model achieved 97,98%[7] accuracy on the test set. Dropout regularization of 0.2 was used after each BiLSTM cell. Experiments included sentences between 5 and 100 characters long. Already for sentences consisting of 5 characters, accuracy reached about 73% and exceeded 95% for the sentences containing at least 14 characters.

We used Keras neural network library with the Tensorflow backend and utilized two Tesla P100 graphic cards to learn the model. The learning phase lasted about 70 hours.

4.3 Results

In addition to the described model, we used two other models for testing on the Twitter data set: a statistical, Naive Bayes model based on the frequency

[7] Different cell sizes were used during experimentation, including 50, 150, 200, 500 dimensional hidden layers, one and two BiLSTM layers. The best results were achieved for 2 layers, each for 200 neurons.

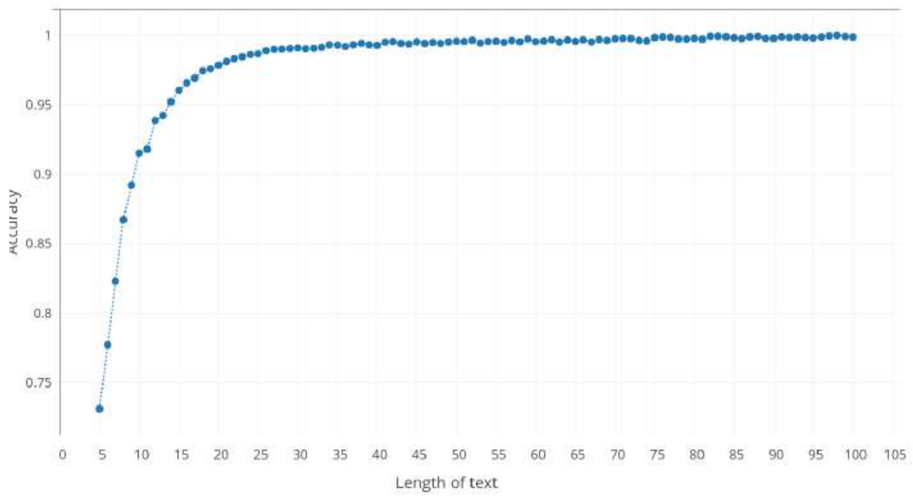

Fig. 2. Accuracy in terms of a sentence length for the test set

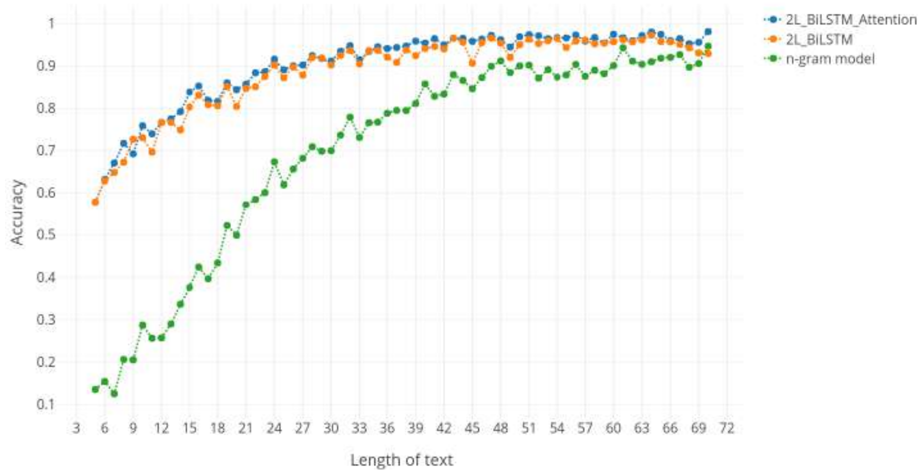

Fig. 3. Accuracy in terms of a length of sentence for the Twitter test set

of occurrences of a n-gram model [8] and a 2-layer BiLSTM model without an Attention layer (the structure and the learning process were the same as in the case of the proposed model). Table 2 shows accuracy achieved by these models. The best result, 91.60%, achieves the BiLSTM model with Attention Mechanism. The difference between the neural models and the statistical model is significant

[8] https://github.com/chbrown/language-detection

(about 15% and 17%) but the difference among the neural models is less than 2%. The detailed results are depicted in the Figure 2.

Table 2. Accuracy measure for the baselines and the proposed model

model	accuracy
n-gram model	74.64%
BiLSTM without Attention	89.69%
BiLSTM with Attention	**91.60%**

Figure 3 outlines the n-gram, the BiLSTM without Attention and the proposed BiLSTM with Attention models evaluation on the Twitter test set. The gap between accuracy scored for short text for the neural networks and the statistical approaches is substantial. For posts consisting at least 24 characters, the neural models achieve about 90% accuracy and keep accuracy between the 90% and 98% for longer posts, whereas the model with Attention Mechanism attains slightly better accuracy for almost all sentence lengths. The statistical model achieves values between 80% and 94% for posts longer than 38 characters.

A more thorough analysis shows that classification mistakes occur more often in the case of the languages pair belonging to the same language family. The most common misclassified pair of languages is Spanish and Portuguese (164 of 7665 posts in Spanish were classified as Portuguese and vice versa 66 of 3643 posts in Portuguese were classified as Spanish). For Spanish, the second most common wrong language prediction is Italian (58 of 7665 posts in Spanish were classified as Italian). For Hindi, the most common misclassified language is Turkish (31 of 3876 posts in Hindi were classified as Turkish).

Except for similarity of languages pair, common reasons for classification mistakes were:

1. short posts or posts build upon proper nouns only, like *oh ok, detroit, ha ok, lady gaga, katy perry, nicki minaj, iiiiiiiiiiidc rihanna*, which occur in many languages
2. posts containing many words written in non formal way, like *retweetonlyi-fyouwantnewfollowers, adoroooo oooooooo, qqqqquuuuuuuuuueeeeeee nooo-jooodaaaaaa mdkcmskck, prettttyyyyyyy huuurrrrrrrtssssss, can u rt this gt please is my dream thanks, puedes dar rt al enlace por favor es mi sue*

5 Conclusion

We have presented the 2-layer BiLSTM neural network with the Attention Mechanism. The network was applied to language identification and achieved 91.60% accuracy on the Twitter posts corpus. The neural model performed better (17% better accuracy) than the statistical baseline model based on a frequency of occurrence of n-grams. The proposed model performs well for short and noisy text and keeps the high performance for longer text. The Attention Mechanism boosts the accuracy of language identification for the examined case.

An improvement over statistical approaches is the effect of basing the inference not on the number of occurrences of the particular letters or n-grams, but on the relations occurring between the letters in a sentence. It turns out that Recurrent Neural Networks are able to learn such characteristic relationships and correctly predict the language of text for short texts. Such relationships are characterized by greater predictive ability than the number of occurrences of specific letters or n-grams for short texts, as demonstrated by our research.

At this point, we think that further improvement can be achieved by increasing the diversity of a training text corpus and adding more advanced embedding layer, like a Convolutional layer, which could be well suited to extend a character embedding with a n-grams embedding.

References

1. Aslam, J.A., Frost, M.: An information-theoretic measure for document similarity. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. pp. 449–450. SIGIR '03, ACM, New York, NY, USA (2003). https://doi.org/10.1145/860435.860545
2. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval. pp. 161–175 (1994)
3. Chang, J.C., Lin, C.: Recurrent-neural-network for language detection on twitter code-switching corpus. CoRR **abs/1412.4314** (2014), http://arxiv.org/abs/1412.4314
4. Damashek, M.: Gauging similarity with n-grams: Language-independent categorization of text. Science **267**(5199), 843–849 (1995), http://gnowledge.sourceforge.net/damashek-ngrams.pdf
5. Du, C., H.L.: Text classification research with attention-based recurrent neural networks. International Journal of Computers **13**(1) (2018). https://doi.org/10.15837/ijccc.2018.1.3142
6. Grefenstette, G.: Comparing two language identification schemes. 3rd International Conference on Statistical Analysis of Textual Data (1995)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735
8. Kim, Y., Jernite, Y., Sontag, D., Rush, A.M.: Character-aware neural language models. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. pp. 2741–2749. AAAI'16, AAAI Press (2016)
9. Kocmi, T., Bojar, O.: Lanidenn: Multilingual language identification on character window. CoRR **abs/1701.03338** (2017), http://arxiv.org/abs/1701.03338

10. Kruengkrai, C., Srichaivattana, P., Sornlertlamvanich, V., Isahara, H.: Language identification based on string kernels. pp. 926 – 929 (11 2005). https://doi.org/10.1109/ISCIT.2005.1567018
11. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: Text classification using string kernels. J. Mach. Learn. Res. **2**, 419–444 (Mar 2002). https://doi.org/10.1162/153244302760200687
12. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1412–1421. Association for Computational Linguistics (2015). https://doi.org/10.18653/v1/D15-1166, http://aclweb.org/anthology/D15-1166
13. Raffel, C., Ellis, D.P.W.: Feed-forward networks with attention can solve some long-term memory problems. CoRR **abs/1512.08756** (2015), http://arxiv.org/abs/1512.08756
14. Selamat, A.: Improved N-grams Approach for Web Page Language Identification, pp. 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24016-4_1, https://doi.org/10.1007/978-3-642-24016-4_1
15. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: ACL (2016)