Language Identification of Similar Languages using Recurrent Neural Networks

Ermelinda Oro[1], Massimo Ruffolo[1] and Mostafa Sheikhalishahi[2]

[1] *Institute for High Performance Computing and Networking, National Research Council (CNR),*
Via P. Bucci 41/C, 87036, Rende (CS), Italy

[2] *Fondazione Bruno Kessler, e-Health Research Unit, Trento, Italy*

Keywords: Language Identification, Word Embedding, Natural Language Processing, Deep Neural Network, Long Short-Term Memory, Recurrent Neural Network.

Abstract: The goal of similar Language IDentification (LID) is to quickly and accurately identify the language of the text. It plays an important role in several Natural Language Processing (NLP) applications where it is frequently used as a pre-processing technique. For example, information retrieval systems use LID as a filtering technique to provide users with documents written only in a given language. Although different approaches to this problem have been proposed, similar language identification, in particular applied to short texts, remains a challenging task in NLP. In this paper, a method that combines word vectors representation and Long Short-Term Memory (LSTM) has been implemented. The experimental evaluation on public and well-known datasets has shown that the proposed method improves accuracy and precision of language identification tasks.

1 INTRODUCTION

Many approaches of Natural Language Processing (NLP), such as part-of-speech taggers and parsers, assume that the language of input texts is already given or recognized by a pre-processing step. Language IDentification (LID) is the task of determining the language of a given input (written or spoken). Research in LID aims to imitate the human ability to identify the language of the input. In literature, different approaches to LID have been presented. But, LID, in particular applied to short text, remains an open issue.

The objective of this paper is to present a LID model, applied to the written text, that results enough effective and accurate to discriminate similar languages, even when it is applied to short texts. The proposed method combines Word2vec (Mikolov et al., 2013) representation and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). The experimental evaluation shows that the proposed method obtains better results compared to approaches presented in the literature.

The main contributions of the paper are:

- Definition of a new LID method that combines the word vector representation (Word2vec) and the classification based on neural network (LSTM RNN).
- Building of a Word2vec representation by using Wikipedia Corpus.
- Creation of a dataset extracting data from Wikipedia for Serbian and Croatian Language, which aren't yet available in literature.
- Experimental evaluation on public datasets in literature.

The rest of this article is organized as follows: Section 2 describes related work, Section 3 shows the proposed model and Section 4 presents the experimental evaluation, Finally, Section 5 concludes the paper.

2 RELATED WORK

In this section, the most recent methods aimed to identify the language in texts is reviewed.

The lack of standardized datasets and evaluation metrics in LID research makes very difficult to contrast the relative effectiveness of the different approaches to a text representation. Results across different datasets are generally not comparable, as a methods efficacy can vary substantially with parameters such as the number of languages considered, the

Oro, E., Ruffolo, M. and Sheikhalishahi, M.
Language Identification of Similar Languages using Recurrent Neural Networks.
DOI: 10.5220/0006678606350640
In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence (ICAART 2018) - Volume 2*, pages 635-640
ISBN: 978-989-758-275-2

relative amounts of training data and the length of the test documents (Han et al., 2011). For this reason, we are particularly interested in related work that makes available datasets and evaluation metrics enabling experimentally comparison.

Malmasi and Dras (Malmasi and Dras, 2015) presented the first experimental study to distinguish between Persian and Dari languages at the sentence level. They used Support Vector Machine (SVM) and n-grams of characters and word to classify languages. For the experimental evaluation, the authors collected textual news from Voice of America website.

Mathur et al. (Mathur et al., 2017) presented a method based on Recurrent Neural Networks (RNNs) and, as feature set, they used word unigrams and character n-grams. For the experimental evaluation, the authors used the dataset DSL 2015[1] (Tan et al., 2014).

Pla and Hurtado (Pla and Hurtado, 2017) applied a language identification method based on SVM to tweets. They used the bag-of-words model to represent each tweet as a feature vector containing the tf-idf factors of selected features. They considered a wide set of features, such as tokens, n-grams, and n-grams of characters. For the evaluation of the implemented system, they used the TweetLID official corpus, which contains multilingual tweets[2] (Zubiaga et al., 2016).

Trieschnigg et al. (Trieschnigg et al., 2012) compared a number of methods to automatic language identification. They used a number of classification methods based on the Nearest Neighbor (NN) and Nearest Prototype (NP) in combination with the cosine similarity metric. To perform the experimental evaluation, they used the Dutch folktale database, a large collection of folktales in primarily Dutch, Frisian and a large variety of Dutch dialects.

Ljubešic and Kranjcic (Ljubešic and Kranjcic, 2014), using discriminative models, handled the problem of distinguishing among similar south-Slavic language such as Bosnian, Croatian, Montenegrin and Serbian languages in Twitter. However, they did not identify the language on the tweet level, but the user level. The tweets collection has been collected with the TweetCat tool, they annotated a subset of 500 users according to language that the user's tweet in. They attempt with the traditional classifiers such as Gaussian Nave Bayes (GNB), K-Nearest Neighbor (KNN), Decision Tree (DT) and linear Support Vector Machine(SVM), as well as classifier ensembles such as Ada-Boost and random forests. They observe that each set of features produces very similar results.

Table 1 summarizes the comparison among the considered related work. Each row of the Table 1 has the reference to the related work. The second column shows the used classification algorithm (such as Nave Bayes, KNN, SVM, Random Forest and Recurrent Neural Network). In the third column is indicated the processed input, i.e., document, sentence or tweet. Documents can have different lengths (both short and long). All approaches use, as extracted features, both character and word n-grams.

Table 1: Comparison of Related Work.

Related Work	Algorithm	Granularity
(Trieschnigg et al., 2012)	Nearest Neighbor	Document
(Pla and Hurtado, 2017)	SVN	Tweets
(Ljubešic and Kranjcic, 2014)	SVM, KNN, RF	Tweets
(Malmasi and Dras, 2015)	Ensemble SVN	Sentence
(Mathur et al., 2017)	RNN	Sentence

Compared to related work, we exploited different ways to represent input features (i.e., character and word n-gram vs word embedding model) and to classify the language (we used LSTM RNN method). In our experiments, we used the datasets exploited in (Malmasi and Dras, 2015; Pla and Hurtado, 2017) and (Mathur et al., 2017) because they are publicly available and we can, in a straightforward way, compare results.

3 PROPOSED MODEL

In this section, we present the proposed method that combines Word2vec with LSTM recurrent neural networks.

Figure 1 illustrates an overview of the proposed LID model.

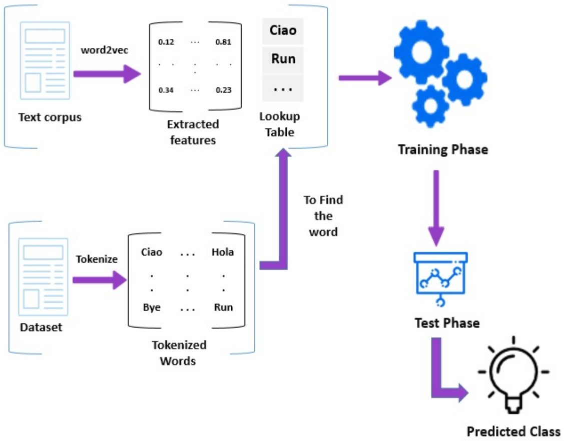

Figure 1: Proposed model.

[1] http://ttg.uni-saarland.de/lt4vardial2015/dsl.html

[2] http://komunitatea.elhuyar.eus/tweetlid/

First, Wikipedia the text corpus of each target language are collected. After the pre-processing, the text is fed to Word2vec that outputs a list of vectors related to each word contained in the input text. Then, a lookup table that matches each vocabulary word of the dataset with its related vector is obtained. During the training phase, the classifier, which corresponds to an LSTM RNN, takes as input the vectors of the dataset. After the training of the classifier, we perform the test phase that takes as input the test set. Finally, the accuracy and precision of the built model are computed.

3.1 Word2vec

The Distributional hypothesis says that words occurring in the same or similar contexts tend to convey similar meaning (Harris, 1954).

There are many approaches to computing semantic similarity between words based on their distribution in a corpus. Word2vec (Mikolov et al., 2013) are model architectures for computing continuous vector representations of words from very large data sets. Such vector representations are capable to find similarity of words not just considering syntactic regularities, but also contextual information. Word2vec takes a large corpus of text as input for training and produces a set of vectors called embeddings, normally having several hundred dimensions, with each unique word in the corpus. Given enough quantity of data, usage, and contexts, this model can make highly accurate guesses about a words meaning based on past appearances. Word2vec produces word embeddings in one of two ways:

- Using context to predict a target word, a method known as ”continuous bag of words” (CBOW)
- Using a word to predict a target context, which is called Skip-gram.

To generate our word embeddings, we chose the CBOW method to train our Word2vec model because the training time is less than Skip-gram and the CBOW architecture works slightly better on the syntactic task than the Skip-gram model.

To feed the word2vec model, we prepared a complete corpus that considers every target language by using Wikipedia. First, we downloaded the wiki dump of each target language available on Wikipedia. Wikipedia provides static dumps of the complete contents of all wiki[3] exported automatically following a rotating export schedule. The contents of these dumps are licensed under the GNU Free. In particular, in April 2017, we obtained XML dumps of Wikipedia with valid ISO 639-1 codes, giving us Wikipedia database exports for six languages (Persian, Spanish, Macedonian, Bulgarian, Bosnian and Croatian) target of this work. We discarded exports that contained less than 50 document. For each language, we randomly selected 40,000 raw pages of at least 500 bytes in length by using the WikiExtractor python script[4].The script removes images, tables, references, and lists. By using another script, we removed links. We removed the stop-words. Then, we tokenized the cleaned text. Finally, we were able to use the obtained corpus to learn the vector representation of words in the different considered languages.

3.2 Long Short-Term Memory

Recurrent Neural Networks (RNNs)(Mikolov et al., 2010) are a special type of neural networks which have an internal state by virtue of a cycle in their hidden units. Therefore, RNNs are able to record temporal dependencies among the input sequence, as opposed to most other machine learning algorithms where the inputs are considered independent of each other. For this reason, they are very well suited to natural language processing tasks and have been successfully used for applications like speech recognition, handwriting recognition (Graves and Schmidhuber, 2009; Graves, 2012; Graves et al., 2013)

Until recently, RNNs were considered very difficult to train because of the problem of exploding or vanishing gradients (Pascanu et al., 2013) which makes it very difficult for them to learn long sequences of input. Few methods like gradient clipping have been proposed to remedy this (Neelakantan et al., 2015). Recently developed architectures of RNNs such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014) were also specifically designed to get around this problem.

In our implementation, we used LSTM neural network. This choice is due to the capability of:

- managing vanishing or exploding gradients problems,
- handling the long time series of data by managing different time steps,
- identifying patterns in sequences of data, such as genomes and text.

In our experiments, we used single hidden layer recurrent neural networks that use Long Short-Term Memory introduced by Graves in (Graves, 2012). Using a good weight initialization brings substantially faster

[3] http://dumps.wikimedia.org/backup-index.html

[4] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

convergence. We use the recently developed weight initialization method which is introduced by Glorot et al. (Glorot and Bengio, 2010).

We implemented our approach by using the Deeplearning4j[5]. It is a Java-based deep learning library developed. It is an open source product made for adaptability in business and released under the Apache 2.0 license. The library provides several tools for text pre-processing, including tokenization, and neural networks implementations. We used this tool also to build the Word2vec and the LSTM RNN models.

4 EXPERIMENTAL EVALUATION

This Section presents the experimental evaluation performed on a new dataset and some well-know datasets. We show that the proposed method improves language identification results.

4.1 Datasets

This Subsection describes datasets used to experimentally evaluate the proposed method. As observed in (Malmasi and Dras, 2015), short sentences are more difficult to classify with respect to longer ones. In fact, may not exist enough distinguishing features if a sentence is too short, and conversely, very long texts will likely have more features that facilitate correct classification. Therefore, to well evaluate our method, we considered four datasets having different lengths of input documents, named: Wikipedia, VOA, DSL 2015, and TweetID. Statistics are shown in Table 2.

Table 2: Datasets statistics.

Dataset	Languages	Train. Size (# of docs)	Test Size (# of docs)	Doc Length (# of words)
Wikipedia	Serbian	3000	1000	5-2500
	Croatian	3000	1000	5-2500
VOA	Persian	3000	1000	5-55
	Dari	3000	1000	5-55
DSL-2015	Bulgarian	18000	2000	22-80
	Macedonian	18000	2000	22-80
TweetID	Spanish	1170	1170	1-25
	Catalan	1190	1170	1-15

4.1.1 Wikipedia Dataset

A first dataset was created by extracting articles from Wikipedia. We considered two similar languages: Serbian and Croatian. As shown in Table 2, we limited the maximum length of each document to 2500 words. We randomly collected a set of 4000 articles per language (3000 training and 1000 evaluation).

4.1.2 VOA Dataset

Malmasi and Dras (Malmasi and Dras, 2015) created a dataset extracting sentences from the Voice of America (VOA) website[6] for Persian and Dari languages. VOA is an international multimedia broadcaster with a service in more than 40 languages. It provides news, information, and cultural programming through the Internet, mobile and social media, radio, and television. The Persian and Dari are similar languages, they are part of the eastern branch of the Indo-European language family and Dari is a low-resourced language. The authors collected sentences in the range of 5-55 tokens in order to maintain a balance between short and long sentences. For this study, as shown in Table 2, we have considered a subset of their dataset, which includes 4000 sentences for each language.

4.1.3 DSL 2015 Dataset

As Mathur et al. (Mathur et al., 2017) done, we used the dataset created for the language competition, named Discriminating between Similar Language (DSL) Shared Task 2015. It includes a set of 20000 instances per language (18000 training and 2000 evaluation) and it was provided for 13 different world languages. In particular, as shown in Table 2, we considered the similar languages Bulgarian and Macedonian.

4.1.4 TweetID Dataset

Zubiaga et al. (Zubiaga et al., 2016) collected a dataset of tweets including seven languages. They exploited the geo-location to retrieve posted from areas of interest. In this work, we considered a subset of Spanish, Catalan tweets. In Table 2 the distribution of train and test dataset is shown in details.

4.2 Results

In this subsection, we show the evaluation results of the presented method that enables to identify the language of each input text. We trained each model using the monolingual training dataset, presented in the previous subsection and verified results considering the test set of the same data sources. Because identifying the languages of each input document is a classification problem, we evaluate results by using the standard notions of accuracy (A) precision (P), recall (R)

[5] https://deeplearning4j.org//

[6] https://www.voanews.com/

and F-score (F). Results are compared with the values and measures asserted in the papers of the related work. Results are summarized in Table 3, where for the approaches existing in literature are shown only results published in the original papers.

Table 3: Comparison between proposed model (LID) and related work, considering different datasets.

Dataset	Model	A	P	R	F
Wikipedia	our approach	97.65	97.74	97.90	97.82
VOA	our approach	98.65	98.89	98.40	98.64
	(Malmasi and Dras, 2015)	96.00			
DLS-2015	our approach	99.50	99.40	99.60	99.50
	(Mathur et al., 2017)	95.12			
TweetID	our approach	88.37	87.40	89.09	88.23
	(Zubiaga et al., 2016)		82.5	74.4	78.2

As shown in Table 3, our proposed method obtained around 97% of accuracy and F-measure considering the collected Wikipedia's documents, which contains documents written in Serbian and Croatian. Our method outperformed the method presented in (Malmasi and Dras, 2015) considering the same dataset that contains sentences in Persian and Dari languages. The improvement is more than 2% of accuracy. Our method outperformed the method presented in (Mathur et al., 2017) considering the same dataset that contains sentences in Bulgarian and Macedonian languages. The improvement is more than 4% of accuracy. Our method performs better than the model that is proposed by (Zubiaga et al., 2016) and the improvement is about 5% using accuracy and is around 12% in F1-Measure as evaluation metrics on the same dataset that contains tweets written in Catalan and Spanish languages.

5 CONCLUSION

In this paper, we presented a method based on neural networks to identify the language of a given document. The method is able to distinguish between similar languages, even when the input documents are short texts, like tweets.

The proposed model has been compared with prior works considering same languages. Experimental evaluation shows that the proposed method obtain better results. However, we intend to more datasets to evaluate our method and work further and deeply on statistical analysis.

There are several modifications that could be tested to improve the proposed method. For example, other features extraction techniques, or other recent neural network based classifier, or more datasets could be used. This work has just shown how the combination of recent deep learning and vector representation techniques allows to getting better results on the problem of language identification of (short) texts.

REFERENCES

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*, volume 385. Springer.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE.

Graves, A. and Schmidhuber, J. (2009). Offline handwriting recognition with multidimensional recurrent neural networks. In *Advances in neural information processing systems*, pages 545–552.

Han, B., Lui, M., and Baldwin, T. (2011). Melbourne language group microblog track report. In *TREC*.

Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ljubešic, N. and Kranjcic, D. (2014). Discriminating between very similar languages among twitter users. In *Proceedings of the Ninth Language Technologies Conference*, pages 90–94.

Malmasi, S. and Dras, M. (2015). Automatic language identification for persian and dari texts. In *Proceedings of PACLING*, pages 59–64.

Mathur, P., Misra, A., and Budur, E. (2017). Lide: Language identification from text documents. *arXiv preprint arXiv:1701.03682*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. (2015). Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318.

Pla, F. and Hurtado, L.-F. (2017). Language identification of multilingual posts from twitter: a case study. *Knowledge and Information Systems*, 51(3):965–989.

Tan, L., Zampieri, M., Ljubešic, N., and Tiedemann, J. (2014). Merging comparable data sources for the discrimination of similar languages: The dsl corpus collection. In *Proceedings of the 7th Workshop on Building and Using Comparable Corpora (BUCC)*, pages 11–15.

Trieschnigg, D., Hiemstra, D., Theune, M., Jong, F., and Meder, T. (2012). An exploration of language identification techniques for the dutch folktale database.

Zubiaga, A., San Vicente, I., Gamallo, P., Pichel, J. R., Alegria, I., Aranberri, N., Ezeiza, A., and Fresno, V. (2016). Tweetlid: a benchmark for tweet language identification. *Language Resources and Evaluation*, 50(4):729–766.