Machine Learning and Neural Networks Tools to Address Noisy Data Issues

Maria Teresa Artese[0000-0001-8453-0807] and Isabella Gagliardi[0000-0002-3706-0993]

IMATI - CNR (National Research Council), Milan, Italy
{teresa,isabella}@mi.imati.cnr.it

Abstract. In this paper, we present tools for addressing noisy keyword issues in digital libraries. Two tasks, language detection and misspelling detection and correction, are addressed using both machine learning and deep learning techniques. To train and validate the models, different datasets were used/created/scraped. Encouraging preliminary results are presented and discussed.

Keywords: Digital Library, Unsupervised Tools, Noisy Data, Tags, Content Based Retrieval.

1 Introduction

Large libraries, which are geographically sparse the territory and require cataloguing to be carried out by several people with different skills and sensibilities, bring to light the need for easy and unsupervised tools for managing noisy data.

In this paper we will define, implement and test tools based on machine learning and deep learning techniques, for the management of noisy keyword (or tag) issues. These tags have been associated in the cataloging phase to books, journals, individual articles or book chapters. The good quality of these data helps immensely researchers, scholars and students in the identification of texts and writings to be studied.

Their main problems are i) the presence of tags expressed in different languages, ii) the extreme specialization of the terms used, iii) the shortness of the terms, and iv) the lack of context.

In this paper, we tackle the problem of processing and exploiting keywords, presenting some preliminary results of tools for the language identification and correction of misspelling. These tools are intended to be used to improve traditional ways of searching and browsing data on the web and offer multimodal search and visualization tools.

The experimentation refers to an Italian scientific library, in order to improve data quality in an unsupervised way. The availability of large datasets of texts, and the possibility of artificially creating them specifically focused, to be used as a basis for statistical analysis, machine learning and deep learning greatly favors their use. Here will be analyzed, compared and commented only results in Italian and English.

The paper is structured as follows: in section 2 we quickly outline the related work, then we describe in full detail the scope we addressed, with the data and the two tasks.

Digital Presentation and Preservation of Cultural and Scientific Heritage. Conference Proceedings. Vol. 11, Sofia, Bulgaria: Institute of Mathematics and Informatics – BAS, 2021. ISSN: 1314-4006, eISSN: 2535-0366

Section 4 presents our approach, possible solutions using machine learning and deep learning, in two languages, Italian and English, with some preliminary results, and a brief discussion to comment on them, including a comparison between the two different models. Conclusions and future work complete the paper.

2 State of the Art

In this paper we will define, develop and test unsupervised tools for natural language processing using machine learning and deep learning approaches. In particular, we deal with the automatic detection of the language in which a text is written, in our case very short, and the automatic detection of writing errors.

Automatic language identification has been studied for over fifty years. Language detection is one of the first activities in the text processing pipeline: for example, automatic translators assume that the input language is known. (Jauhiainen, 2019) provides an extensive survey of the features and methods used in the literature. Recently ensemble methods are studied for their ability to take the best results from different methods and integrate them (Mukherjee, 2020). In 2019, one of SemEval tasks has been the use of ensemble machine learning to detect hate speech (Ramakrishnan, 2019).

In recent years, deep learning models such as RNN and LSTM, and word embedding models such as word2vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) (Mikolov, et al., 2020) or BERT (Devlin, 2018) have greatly improved text comprehension and management. (Goldberg, 2015) surveys neural network models from the perspective of natural language processing research, in an attempt to bring natural-language researchers up to speed with the neural techniques. Different neural network methods for the automatic language identification have been proposed (Simões, 2014), (Botha, 2012), (Lopez-Moreno, 2014). In (Hládek, 2020), the authors provide a systematic overview of the approaches developed so far. To account for context, LSTM (Cho, 2014) and sequence2sequence (Sutskever, 2014) architectures were used for automatic error correction. The lack of resources for specific languages requires special consideration, as in (Etoori, 2018).

3 The Problem

Every library of an academic institution tries to balance the work of its catalogers to maximize the usability of its catalogs to the intended users. Especially in the scientific field, knowledge and research activities move at a speed unthinkable just a few decades ago: the lists from which to draw keywords are not always available or up-to-date.

Therefore, keywords to be associated with books, articles, proceedings, etc. can be chosen from predefined lists, or written down. The tools presented in this article aim to improve quality data, referring to the latter mode.

The tools have been designed and defined to, automatically and in an unsupervised way:

- identify the language of the tags: this is particularly useful when the user interface is offered in different languages, for users of different nationalities/languages.
- to identify and correct misspelling: this is essential in order not to "miss" search results that may be important or critical.

3.1 Data

The data of this experiment are extracted from a library of an Italian research institution and refer to books, journals, reports for a total of more than 300000 objects. The cataloguing has been carried out since the late 60's, and strictly follows the Reicat 1 rules adopted in Italy. For the specific purpose, only the keywords have been used, without any reference either to the titles to which they refer, or to the subject used, even when known. Usually, in natural language processing (NLP) tasks such as keyword extraction, or automatic summarization, the first operation is preprocessing, whose purpose is cleaning, lemmatizing, stemming, and pos tagging the set of terms, single or compound words, which can be labeled as keywords or key phrases, in the next steps. In our case, the dataset is already composed only of simple terms, so the pre-processing step may be unnecessary or optionally consist only of lemmatization and pos tagging.

3.2 Task 1: Automatic Language Identification

Keywords associated with collections of objects can be in the same language in which the object itself is described or catalogued, or in the language of the country in which it is catalogued. Therefore, when we are dealing with multilingual collections, for example, books and journals of a library, or scientific production of a university or research institute, the tags can be indifferently in English, Italian, French, ... Automatic language identification, using statistical techniques, Machine Learning or Deep Learning methods, is therefore necessary. In the following, two methods are presented and compared, detailing them.

3.3 Task 2: Misspelling Identification and Correction

This work is designed for specialized keywords in research areas. Tags can be extremely specific and focused and not fit into common lexicons. Because of the extreme specificity of terms, and also their volatility and updating, using standard methods such as presence in WordNet[2] (Fellbaum, 2010) or term lists alone is not sufficient to determine the presence of misspellings. In this paper, both machine learning and deep learning based solutions are presented. Machine learning methods lead to the identification of the presence of an error and suggestions for its correction, through statistical and probability functions. The deep learning method, trained on different training sets, identifies and corrects errors in a single step. We created a synthetic dataset of noisy and

[1] https://norme.iccu.sbn.it/index.php?title=Reicat

[2] https://wordnet.princeton.edu/

correct words for both Italian and English, collecting highly probable spelling errors and inducing noise in the clean corpus. To obtain accurate results, training requires many epochs and large data sets.

Table 1: Steps of the proposed approach

Task 1: Language identification
- Datasets preparation (Wikipedia pages)
- Compute word embedding trained on ArXiv dataset
#ML approach
- For each tag in n tags:
- apply language identification methods
- voting system on results of the statistical/ML methods
#Neural Network approach
- using the features identified, an ANN has been defined, trained and validated on the prepared dataset.
- the trained ANN has then been applied to library tag dataset.
Task 2: Misspelling detection
- Datasets preparation (ArXiv, Wikipedia, word embedding models- pretrained and trained on ArXiv – artificially created datasets with misspellings)
#ML approach
is the tag correct?
- For each tag in n tags:
- test several identification strategies
- voting system on results of the methods, with a confidence score
tag substitution suggestion: correction strategies
- several strategies: Automatic corrections/google first results/split
- voting system (ensemble)
#Neural Network approach
a single step to identify misspelling and suggest correction
- an ANN has been defined (seq2seq), trained and validated on the datasets specifically created.

4 Our Approach

Our approach has been implemented, according to the steps defined in Table 1, described in depth in their computational aspects.

Each task has been trained and evaluated on separate datasets, which will be described in more detail below. No pre-processing was done in this experiment. The tags, exactly as entered by the cataloguers/researchers were used.

4.1 Task 1: Automatic Language Identification

Datasets and models. For training and testing the models, we scraped web pages from Wikipedia, starting from the root categories, available in all the languages we are interested in, e.g. culture, history, technology, food, etc. For each category, the scraper tool extracted all the pages of that category, in a recursive way, for a depth of k level (for this experiment, k has been set to 2, a balanced trade-off between the total number of

documents to be processed and the variety of terms to consider). The language we are interested in are: English, French, German, Italian and Spanish. Here we reported results for English and Italian. The neural network model has then been trained on the Wikipedia dataset split in training/test, and validated both on the training set/ test set and on a set of data taken from the library we are working on.

Machine Learning Approach. For the automatic language identification, we used statistical models or pre-trained machine learning methods, to be used as a comparison for the evaluation of the Deep learning method. In particular off-the-shelf methods, polyglot[3], FastText[4] and Langid[5] were used, as implemented in python. A majority voting mechanism led to the final identification of the language. In table 2 the accuracy of the ML methods tested are reported, and the majority voting (Ensemble – in grey) mechanisms. Results are reported for both the Wikipedia dataset and library dataset. The methods were applied as is, without fine tuning or retraining on the specific data used, which could have brought further improvement. It can be seen that the voting mechanism greatly improves the accuracy of the results.

Table 2: Accuracy values for ML methods

Methods	Wikipedia dataset	Library dataset
Langid	0.778	0.704
Polyglot	0.726	0.861
Fasttext	0.798	0.867
Ensemble	0.852	0.940

Deep Learning Approach. There are several different deep learning approaches to language identification, such as reported in (Simões, 2014), (Lopez-Moreno, 2014). Here we will use a simple RNN, using two different features: character n-grams and single words. The Neural Network model is constituted by 3 layers fully connected, and at the end, classify each input word as belonging to a language class.

As in the ML approach, also in this case we have to transform the dataset into vectors, to be understood by Neural Network model. We need to identify these features that better characterize our data, extract them and create a feature matrix. Here we tested two features:

- Single words and
- character n-grams which are sets of n consecutive characters: in this case n=3. This is a similar approach to a bag-of-words model except we are using characters and not words.

Then a feature matrix, one for each feature selected, will be created, based on the occurrences of each character trigram/word in the dataset. In the training phase for all the cases tested there is no overlap on the features in the various languages. This obviously makes that the model succeeds to discriminate very well the various languages.

[3] https://github.com/saffsd/polyglot
[4] https://fasttext.cc/docs/en/support.html
[5] https://github.com/saffsd/langid.py

Table 3 reports the results, in term of loss/ accuracy, obtained applying the model comparing the two features, and different number of features. The greater the number of features extracted, the better the results that can be obtained. Extracting the n most frequent elements, with n=1200, 600 and 400, as the number of n increases, the number of features extracted from the data increases, thus allowing to obtain a higher accuracy in the results. Here are reported the results obtained with 10 epochs. 50 epochs were tested, but the accuracy of the results did not change, while the computational time greatly increased. The results obtained using trigrams are better than the results obtained on words. Comparing then with the pretrained methods it can be noticed that the results of the methods using deep learning outperform the ml methods, on all tests.

Table 3: Accuracy values for NN methods

feature	no. features	features extracted	epochs	dataset	Loss / Accuracy
trigrams	1200	1693	10	Train	Loss: 0.008 – accuracy: 0.996
				Test	Loss: 0.245 – accuracy: 0.966
				Library	Loss: 0.638 – accuracy: 0.955
	600	894	10	Train	Loss: 0.024 – accuracy: 0.989
				Test	Loss: 0.237 – accuracy: 0.964
				Library	Loss: 0.273 – accuracy: 0.947
	400	612	10	Train	Loss: 0.034 – accuracy: 0.984
				Test	Loss: 0.217 – accuracy: 0.958
				Library	Loss: 0.320 – accuracy: 0.938
words	1200	2329	10	Train	Loss: 0.139 – accuracy: 0.940
				Test	Loss: 0.236 – accuracy: 0.933
				Library	Loss: 0.256 – accuracy: 0.974
	600	1175	10	Train	Loss: 0.166 – accuracy: 0.930
				Test	Loss: 0.196 – accuracy: 0.925
				Library	Loss: 0.256 – accuracy 0.975
	400	785	10	Train	Loss: 0.177 – accuracy: 0.925
				Test	Loss: 0.195 – accuracy: 0.921
				Library	Loss: 0.258 – accuracy: 0.974

4.2 Task 2: Misspelling Identification and Correction

Due to the specific scope of the experiment, identifying misspelled words was a particularly difficult task due to: i) different languages; ii) very specialized scientific terms; iii) presence of proper nouns/project names.

Datasets and Models. We used different datasets for statistical purposes in ML models and for training/test.

Statistical/ ML approach. For the terms that were recognized in English, the dataset of all papers in the ArXiv repository was used (Cornell_University, 2020) (about 1,700,000 papers). This was done to ensure the presence of specialized scientific terms that might be excluded from a standard vocabulary. For terms identified as being in

Italian, a more standard dataset was used, the October 2020 Italian Wikipedia dump[6], containing more than one million articles. In this case, we keep only the title and abstract of each article.

Another suggestion of the presence of misspellings is whether the term is classified as oov (out of vocabulary) in the word embedding model. To minimize the likelihood that a correct but infrequent term, because it belongs to a specific context or is freshly introduced, would be classified as oov, we used either pre-trained word embedding or specifically trained word embedding. In a word embedding model, each word is represented as a real-valued vector in a predefined vector space, and the vector values are learned in a way that resembles a neural network, so the technique is often included in the field of deep learning. Several models have been developed since 2013 when the first models appeared: here we use word2vec, which is one of the most widely used techniques for learning word embedding using a shallow neural network (Mikolov, et al., 2020), (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013).

Deep Learning approach. Datasets have been artificially created, adding misspelling.

We used two starting datasets and two misspelling algorithms. For starting datasets, we extracted single terms as in the following:

- from ArXiv dataset for English and title and abstract from Italian Wikipedia (the same used above): this dataset is called General;
- from library keywords, both in Italian and in English: this dataset is called Library;
- mixed data from General and Library.

The two misspelling algorithms mimic the most frequent errors that are introduced when writing:

- simple changes to a word, such as a deletion (removing a letter), a transposition (swapping two adjacent letters), a substitution (changing one letter for another), or an insertion (adding a letter). This algorithm is called edit1, and is based on the correction algorithm of Peter Norvig[7]
- replacement of a letter with another nearby in the keyboard or deletion of a letter: this algorithm is called nearby.

The same training/test datasets have been used to evaluate ML approach.

Machine Learning approach. In misspelling identification step, several strategies were fielded, to be integrated, since individually none proved sufficient in the specific case. All these strategies indicate hints of the correctness of the tags:

- Presence of the tag in the word embedding model adopted;
- presence of "as is" tags (both compound and as single components) the ArXiv dataset /Italian Wikipedia;
- presence "as is" in the first results of Google;
- presence in synsets of WordNet/MultiWordNet (the multilingual version of WN).

[6] https://dumps.wikimedia.org/backup-index.html

[7] http://norvig.com/spell-correct.html

Once the presence of misspellings is detected with a certain degree of confidence, the second step of the approach is applied, using statistical and/or Machine learning methods to assess the probability of replacing one term with another. It also uses the first results of a Google search or the "do you mean" function to propose a correction: it proposes several tools for automatic correction as follows:

- automatic correction of the error, due, e.g., to the inversion of two characters, to the substitution of two close characters in the keyboard, … (Norvig algorithm)
- Google search first n results, with n =1;
- split terms that have been proved to be pasted (lack of space)

A voting system (ensemble) has been defined, which associates to each term (suggestion of correction) a degree of confidence. Table 4 reports the results for each step of the automatic correction tools, and the final ensemble (in grey) accuracy values. Due to the method used to introduce errors, edit1 misspelling algorithm yields better results than those nearby.

Table 4: Accuracy values for ML approach for misspelling correction

	Nearby Misspelling algorithm			Edit1 Misspelling algorithm		
Datasets	General + library	General	Library	General + library	General	Library
Presence as is	0.249	0.266	0.226	0.256	0.266	0.233
Autom. correction	0.794	0.723	0.863	0.822	0.741	0.906
Google first result	0.180	0.145	0.219	0.234	0.204	0.269
Ensemble	0.881	0.872	0.893	0.919	0.903	0.928

Deep Learning approach. The basic idea has been to use a neural network designed to transform an input into an output, adapted to natural language, thus capable of remembering the various states of the input. The network chosen has been sequence2sequence, which have been widely used for machine translation purposes. This network, also called NN Encoder–Decoder, consists of two recurrent neural networks (RNN) that act as an encoder and a decoder pair. The two networks are trained jointly to maximize the conditional probability of the target sequence given a source sequence. Additionally, the hidden units have been inserted to improve both the memory capacity and the ease of training. LSTM (Long Short Term Memory) layers have been used. The model is based on character 1-gram model. The unique input tokens are 33, while the outputs are 35.

Table 5 shows the loss and accuracy of the network on the augmented data. The model generalizes quite well, keeping the accuracy high, but losing a few points when using the other dataset. Analyzing the results of both methods, we can see that the datasets obtained applying edit1 are corrected in a better way, compared to nearby. Comparing the results, we can see that the neural networks outperform the voting mechanism of ML on all tests.

The experimental setup has been implemented in Python 3.7, using standard packages like Numpy, Matplotlib, Pandas and other more specific ones for processing of textual data such as NLTK, Treetagger, Gensim, Sklearn, Pytorch, Tensorflow, Keras, together with some experimental packages in GitHub.

Table 5: Rsults for NN based misspelling correction

Training dataset	dataset	Loss / Accuracy/ Nearby misspelling	Loss / Accuracy Edit1 misspelling
General + Library	Train	L: 0.008- A: 0.998	L: 0.002- A: 0.999
	General	L: 0.051- A 0.998	L: 0.005- A: 0.998
	Library	L: 0.008 - A 0.998	L: 0.008 – A: 0.998
General	Train	L: 0.004 - A: 0.999	L: 0.002 – A: 0.995
	General	L: 0.066 - A: 0.988	L: 0.057 – A: 0.989
	Library	L: 0.172 - A: 0.976	L: 0.146 – A: 0.977
Library	Train	L: 0.001 - A: 0.999	L: 0.001 - A: 0.996
	General	L: 1.558- A: 0.814	L: 1.453 A: 0.8456
	Library	L: 0.012 - A: 0.998	L: 0.009 - A: 0.998

5 Conclusion and Future Works

In this paper, we presented possible tools for handling noisy keyword issues in digital libraries. These tools are designed for librarians who find themselves managing data from different sources, and who require cleaning based on the language in which it is written. The two tasks, language detection and typo detection and correction, were addressed with both machine learning and deep learning techniques. In order to train and validate the models, different datasets were used/created/scraped, both in a single language (Italian/English) and mixed. The mixed datasets were the basis for the language detection models. They were also used to test the other models in task 2, and to train the deep learning model. In addition, 4 datasets were artificially constructed for automatic error correction, based on two different error models.

We plan to include these tools both at the cataloging stage, to help catalogers detect errors early, and to facilitate the use of library archives in the presence of dirty data. Often researchers, students, scholars use keywords to search for items of interest. The presence of errors in this information could penalize them greatly.

The work is still in progress. The preliminary results have been discussed with the cataloguers in order to evaluate them, both qualitatively and quantitatively. They have been received very positively. Some problems have already emerged in the automatic detection of errors, in the creation of datasets to train the model. There are cases, especially when tags refer to very specific instances, e.g., on-going projects, that it is impossible to automatically determine whether the tag is correct or not.

The work you plan to do involves:

* Extend managed languages to French, German, Spanish and consider including language extensions to languages that use different alphabets, such as Chinese (simplified), Japanese, Russian;
* consider not only syntactic errors, but also semantic ones. For this it will be necessary to consider the context in which the keywords are used. If available, the title and an abstract will also be used;

* Consider organizing in clusters or groups, those keywords that are semantically equivalent.

References

Botha, G. R. (2012). Factors that affect the accuracy of text-based language identification. *Computer Speech & Language*, 307-320.

Cho, K. V. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *ArXiv preprint*.

Cornell_University. (2020). *ArXiv dataset.* Tratto da Kaggle: https://www.kaggle.com/Cornell-University/arxiv

Devlin, J. M.-W. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*.

Etoori, P. M. (2018). Automatic spelling correction for resource-scarce languages using deep learning. *Proceedings of ACL 2018, Student Research Workshop.*

Fellbaum, C. (2010). WordNet. In *Theory and applications of ontology: computer applications.* (p. 231-243). Springer.

Goldberg, Y. (2015). A Primer on Neural Network Models for Natural Language. *ArXiv Preprint*. Tratto da https://arxiv.org/pdf/1510.00726.pdf

Hládek, D. J. (2020). Survey of Automatic Spelling Correc-tion. *Electronics* .

Jauhiainen, T. L. (2019). Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 675-782.

Lopez-Moreno, I. G.-D.-R. (2014). Automatic language identification using deep neural networks. . *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, (p. 5337-5341).

Mikolov, T., Chen, K., Corrado, G., Dean, J., Sutskever, L., & Zweig, G. (2020, 3 27). *Tool for compu-ting continuous distributed representations of words: word2vec.* Tratto da google: https://code.google.com/p/word2vec

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *In Advances in neural information pro-cessing systems 26*, 3111-3119.

Mukherjee, H. D. (2020). An ensemble learning-based language identification system. *Computational Advancement in Comm. Circuits and Systems*, 129-138.

Ramakrishnan, M. Z. (2019). UVA wahoos at SemEval-2019 task 6: Hate speech identification using ensemble machine learning. *Proceedings of the 13th International Workshop on Semantic Evaluation*, (p. 806-811).

Simões, A. A. (2014). Language Identification: a Neural Network Approach. *3rd Symposium on Languages, Applications and Technologies.*

Sutskever, I. V. (2014). Sequence to sequence learning with neural networks. . *ArXiv preprint*.

Received: June 28, 2021
Reviewed: July 15, 2021
Finally Accepted: July 23, 2021