PROCEEDINGS OF SCIENCE

Authorship recognition and disambiguation of scientific papers using a neural networks approach

Sebastiano Fabio Schifano[*]
University of Ferrara and INFN (ITALY)
E-mail: sebastiano.schifano@unife.it

Tommaso Sgarbanti
University of Ferrara (ITALY)
E-mail: tommaso.sgarbanti@student.unife.it

Luca Tomassetti
University of Ferrara and INFN (ITALY)
E-mail: luca.tomassetti@unife.it

Authorship recognition and author names disambiguation are main issues affecting the quality and reliability of bibliographic records retrieved from digital libraries, such as Web of Science, Scopus, Google Scholar and many others. So far, these problems have been faced using methods mainly based on text-pattern-recognition for specific datasets, with high-level degree of errors.

In this paper, we propose a different approach using neural networks to learn features automatically for solving authorship recognition and disambiguation of author names. The network learns for each author the set of co-writers, and from this information recovers authorship of papers. In addition, the network can be trained taking into account other features, such as author affiliations, keywords, projects and research areas.

The network has been developed using the TensorFlow framework, and run on recent Nvidia GPUs and multi-core Intel CPUs. Test datasets have been selected from records of Scopus digital library, for several groups of authors working in the fields of computer science, environmental science and physics. The proposed methods achieves accuracies above 99% in authorship recognition, and is able to effectively disambiguate homonyms.

We have taken into account several network parameters, such as training-set and batch size, number of levels and hidden units, weights initialization, back-propagation algorithms, and analyzed also their impact on accuracy of results. This approach can be easily extended to any dataset and any bibliographic records provider.

International Symposium on Grids and Clouds (ISGC) 2018 in conjunction with Frontiers in Computational Drug Discovery
16-23 March 2018
Academia Sinica, Taipei, Taiwan

[*] Speaker.

Citation Id	Citation
c_1	(r_1) E. Calore, (r_2) . . . , (r_3) . . . , (r_4) S.F. Schifano, (r_5) . . . (2017) doi:10.1002/cpe.3862
c_2	(r_6) S. Chiappini, (r_7) F. Schifano (2018) doi:10.1097/JCP.0000000000000814
c_3	(r_8) S. Chessa, (r_9) . . . , (r_{10}) . . . , (r_{11}) . . . , (r_{12}) F. Schifano, (r_{13}) . . . (2002) doi:10.1049/ip-cdt:20020808

Table 1: Example of *synonyms* and *homonyms*

1. Introduction and Background

Several digital libraries (DL), like Scopus, Web Of Science, Google Scholar and many others, provide today services that facilitate the finding of research publications. DL are increasingly becoming the primary source of information for academic and research communities; they store millions of bibliographic records, and allow to easily find publications reporting results of several research areas. The information stored by DL are not used only by researchers, but also by, for example, institutional agencies on the award of grants or on decisions for funding research project. They are also used to monitor the quality of research performed. All this requires that the contents of DL is reliable. However, the information associated to the bibliographic records may be affected by errors depending by several factors: data-entry errors, lack of standards citation formats errors in collecting citations, ambiguous author names, etc. Among all these sources of errors, *author name ambiguity* is one of the most challenging because of its inherent complexity. Specifically, authorship ambiguity is a problem affecting bibliographic record sets stored in several digital libraries, where the same author may appear under distinct names (*synonyms*), or distinct authors may have similar names (*homonyms*). Both, *synonyms* and *homonyms*, decreases the quality and reliability of information of DL.

To describe better the problem let consider the three publication records – or *citations* – c_i ($1 \leq i \leq 3$) reported in Tab.1 and extracted from *Scopus DL*. In each citation, the authors are identified by names – or *reference* – r_j ($1 \leq j \leq 13$). In this example, references r_4 and r_{12} are examples of *synonyms* because both refer to *Sebastiano Fabio Schifano* from University of Ferrara and INFN, while references r_7 and r_{12} are examples of *homonyms* because r_7 refers *Fabrizio Schifano* from University of Hertfordshire. In a more formal way the problem can be described as follow: let $\mathscr{C} = \{c_1, c_2, c_3, \ldots\}$ a set of citations, the task of *disambiguation* is to define a function that partition the data-set of references $\mathscr{D} = \{r_1, r_2, r_3 \ldots\}$ into n sets $\{\mathscr{S}_1, \ldots, \mathscr{S}_n\}$, so that each partition $\mathscr{S}_i$ contains all the references to a same author [1].

The challenging problem of name disambiguation has been address by several studied using different approaches. Broadly speaking, the most common methods used to address the problem of names disambiguation are based on *author grouping* (see e.g. [2]) and *author assignment* (see e.g. [3]). The first method groups the references to the same author applying a *similarity function* to the names. The similarity function can be *predefined*, for example the *Levenshtein* function distance counting the number of operations to transform one string into the other, or *learned* using a supervised machine learning technique, or *extracted* from the relationships among the authors and coauthors usually represented as a graph. The similarity function is then used to group (cluster) the references to the same author. The latter approach, assigns directly the references to their respective authors using either a *supervised classification* which defines a disambiguation function which relates the reference to the correct author, or a *clustering technique* using probabilistic approaches. For a more deep and complete overview of these methods see [4, 1].

In this paper we use an approach based on *Author Assignment*, where the disambiguation function is defined using *supervised classification* based on *Artificial Neural Network* (ANN). The network receives a set $\mathscr{D}$ of references to authors for which the correct authorship is known, and – after an appropriate training – produces a disambiguation function that relates the reference to the correct author.

The remainder of the paper is organized as follow: the next section gives an overview of ANN, the section 3 describes the steps we have done to trains the ANN, the section 4 presents the performance results achieved, and finally the section 5 addresses some concluding remarks.

2. Artificial Neural Networks

Artificial neural networks (ANNs) are discussed and described in details in several papers, books and text-books [5, 6, 7]. In brief, ANNs are computer programs designed to simulate the way in which the human brain processes information. ANNs gather their knowledge by detecting the patterns and relationships in data and learn (or are trained) through experience, not from programming. An ANN is formed from several single units, called processing elements (PE) or artificial neurons, inter-connected with coefficients – weights –, which constitute the neural structure, eventually organized in layers. The power of neural computations comes from connecting neurons in a network. Each PE has weighted inputs, transfer function and one output. The behavior of a neural network is determined by the transfer functions of its neurons, by the learning rule, and by the architecture itself. The weights are the adjustable parameters and, in that sense, a neural network is a parameterized system. The weighed sum of the inputs constitutes the activation of the neuron. The activation signal is passed through transfer function to produce a single output of the neuron. Transfer function introduces non-linearity to the network. During the training, the inter-unit connections are optimized in order to minimize the error, obtaining a gradual increment of accuracy. Once the network is trained and tested it can be given new input information to predict the output on previously unseen/unknown cases.

A schematic example of such an ANN is depicted in Fig. 1, where an interconnected group of nodes, representing each an artificial neuron is shown; the example shows the input (left), the output (right) and one hidden (center) layers; each arrow represents a connection from the output of one artificial neuron to the input of another. The input data of the problem are fed to the network; each node receives inputs from the previous layer, weights it on the basis of its importance, computing a weighted sum of the inputs and adding an additional input called bias with weight b to properly tune its output value; a non-linear function called activation function (typically a sigmoid) is then applied to the weighted-sum and bias in order to produce the output value, which is in turn fed to the next layer or becomes the result of the network.

Given n input features and m output classifications, the linear combination of inputs of neuron k is given by $u_k = \sum_{j=1}^{n} x_j \cdot w_{k_j}$, where x_j is the j feature of the input and w_k the weight vector of neuron k – with w_{k_j} the j element of the vector w_k– and the output of neuron k is $y_k = f(u_k + b_k)$, with b_k the bias associated to neuron k. f is the activation function, usually the sigmoid $f(x) = \frac{1}{1+e^{-x}}$.

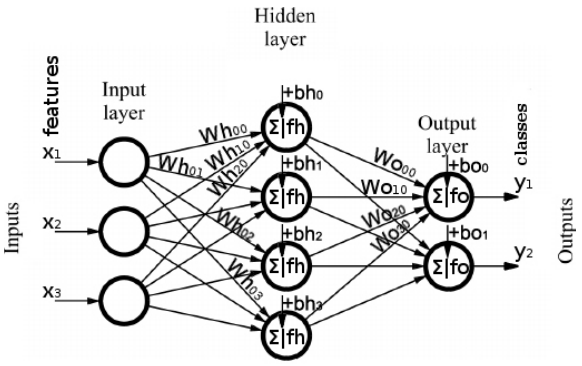

Figure 1: Example of Artificial Neural Network with input, output and one hidden layer.

During the training, the weights are initialized to –small – random values and the bias to zero. Then each input $d \in \mathscr{D}$ from the training set is forwarded from one layer to the others until it reaches the output layer; in this phase, the network computes the output o and the error ε corresponding to the input d. The error ε is then propagated backwards changing the values of weights and bias in order to reduce the error itself.

Neural networks can be trained using different approaches: online, when inputs are processed sequentially and weight and bias corrections are computed for each output; batch, when all inputs are processed in batch and correction happens only once; mini-batch, when the training-set is divided and processed in small batches and the weight and bias correction is computed for each batch. One epoch defines when the entire data-set is processed forward and backward through the neural network. In order to reach high precisions several epochs may be necessary to train the network.

2.1 Performance measurements and metrics

In order to assess the quality of the output produced by the network after the training phase on a validation-set, the results are summarized according to the table of confusion reported in Table 2. For authorship recognition and disambiguation purposes, the meaning of actual and predicted cases and their relationships are the following:

TP – True Positive: the network identifies correctly an author cited in the document;

TN – True Negative: the network does not identify an author that was not cited in the document;

FP – False Positive: the network identifies an author that was not cited in the document; *Type I error*;

FN – False Negative: the network does not identify an author that was cited in the document; *Type II error*.

The following metrics can be used to quantify the performances of the network predictions:

Precision: (or positive predictive value) $\frac{\text{TP}}{\text{TP+FP}}$, the fraction of True Positive among all that predicted positive (i.e. out of the number of authors identified by the network, the fraction of those that are really cited in the documents);

		Actual Cases	
		Positive	Negative
Predicted Cases	Positive	TP	FP (α)
	Negative	FN (β)	TN

Table 2: Table of confusion for the whole authorship recognition and disambiguation problem; In the predicted cases, Positive and Negative correspond to recognized and missed author, respectively. In the actual cases, Positive and Negative correspond to present and absent author, respectively.

Recall: (or sensitivity or true positive rate) $\frac{\mathrm{TP}}{\mathrm{TP+FN}}$, the fraction of True Positive predictions among all actual positive cases (i.e. out of the number of authors that are really cited in document, the percentage of those identified by the network);

Accuracy: $\frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$, the fraction of cases correctly classified (i.e. the number of authors correctly identified in the documents);

F1 score: $2 \cdot \frac{\mathrm{precision \cdot recall}}{\mathrm{precision + recall}} = \frac{\mathrm{2TP}}{\mathrm{2TP+FP+FN}}$, the harmonic mean of precision and recall.

3. Design of Neural Network for Authorship Disambiguation

For our application we have used an ANN with one hidden layer [8]. To properly design and tune the network parameters, we have then performed several test using a data-set $\mathscr{D}$ including 3463 documents with 14574 references. For the tests we target to recognize and disambiguate 28 authors. In the following we show the performance of the network changing several parameters: fraction of input data-set used for the training of the network, the size of batches, the initialization values of weights and the number of neurons to use for the hidden layer. For each parameter we select the values that allow to reach a good level of accuracy – e.g. $> 98\%$ – with a small number of epochs.

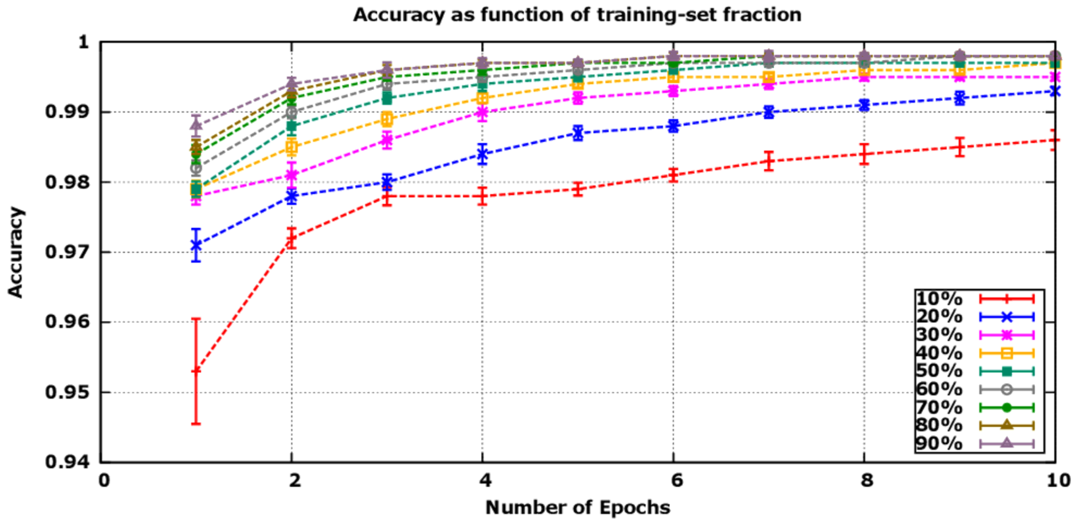

Figure 2: Accuracy as function of training-set fraction used for the training.

In Fig. 2 we report the accuracy as function of the fraction of the input data-set $\mathscr{D}$ we use as training set for the network. The accuracy increase using large fractions of input data-set as training-set, and a value $> 98\%$ is quickly achieved with small number of epochs using a fraction larger than 50%.

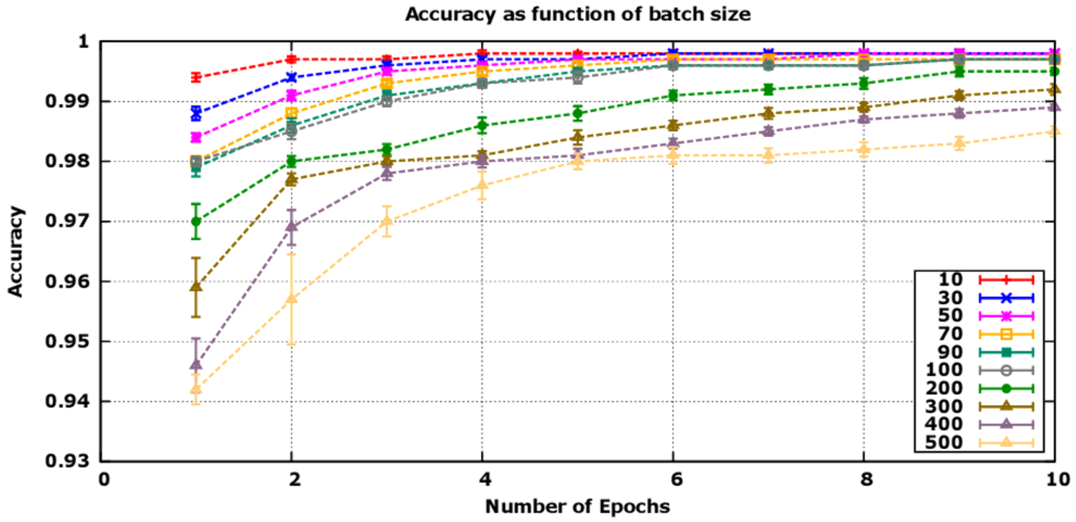

Figure 3: Accuracy as function of batch size.

In Fig. 3 is reported the performance of the network as function of the batch size. As expected, small batches allows to reach high accuracy very fast – one epoch – but as we discussed this may make the training process very slow. From the plot we see that a good comprise can be to use batches of size 50-100 which allow to reach quickly an accuracy $> 98\%$.

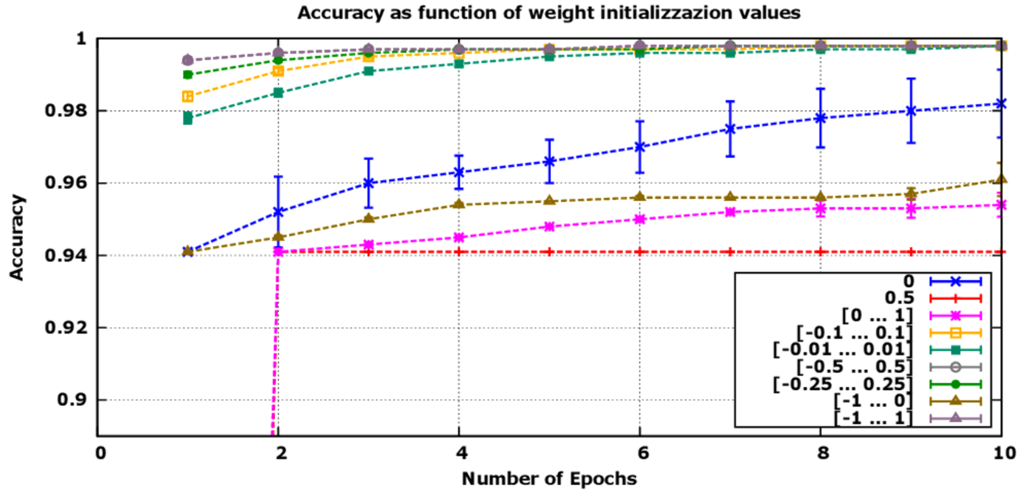

Figure 4: Accuracy as function of weight initialization values.

Fig. 4 reports the accuracy of the ANN using different initialization values of weights. We see that starting with all weights either ≥ 0 or ≤ 0, lines blue, red, purple and brown, the accuracy is low and increases slowly or remains constant. A better result is achieved using an initialization of weights with random values uniform distributed on a range around zero. For example using values in the range $[-0.1\ldots+0.1]$ allows to quickly achieve an accuracy $> 98\%$.

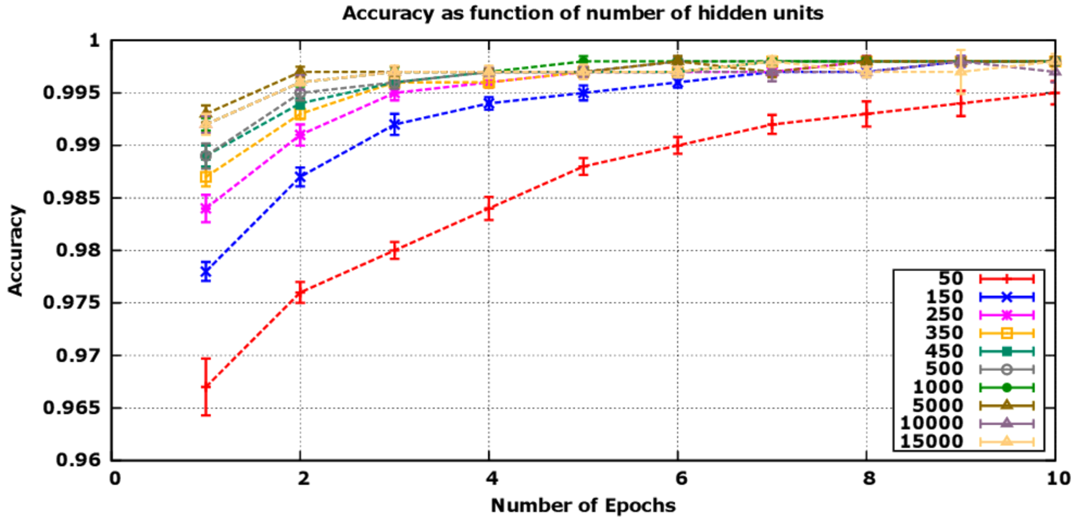

Figure 5: : Accuracy as function of number of hidden layers.

Finally, Fig.5 reports the accuracy as function of the number of neurons used for the hidden layer. The accuracy increases with larger number of epochs, and a value > 98 is achieved using > 250 neurons.

From the results of the calibration tests, we have then decided to use 70% of input data-set $\mathscr{D}$ as training-set and 30% as validation set, batch size of 50, weights w initialized with random value uniformly distributed in the range $[-0.1\ldots+0.1]$, and 500 neurons for the hidden layer.

4. Results

After the tuning of network parameters, as described in the previous section 3, the performances of the ANN have been measured against the validation-set, consisting of 30% of the entire data-set, namely 1035 documents.

Table 3 shows a summary of the results, that includes the number of documents used for training, the metrics and the confusion matrix, for each of the 28 cases/authors of interest. The average Accuracy is 99.9%, the average Recall is 95.4% and the average Precision is 98.6% demonstrating the effectiveness of our approach. Only a few False Positive or False Negative occurrences in a limited number of authors have been registered. The first case, which occurs when the network identifies an author that was not cited in the document – a Type I error, a false hit – can be easily cured by post-processing the network output using the full author list. The latter, which occurs when the network does not identify an author that was cited in the document – a Type II error, a miss – is more problematic and is linked to the presence of several authorship patterns for the given author and thus requires an extended training-set if available or the use of additional features in the input to be cured.

It is worth to note that several cases (Elisa Fioravanti for instance) show a Recall of 100%, while one case (Franco Mantovani) shows a Recall of 50% only. The latter suffers of a limited

Author	Training	$\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$	$\frac{\text{TP}}{\text{TP+FN}}$	$\frac{\text{TP}}{\text{TP+FP}}$	True	False	True	False
	Documents	Accuracy	Recall	Precision	Positive	Positive	Negative	Negative
Alberti Marco	37	0.999	0.933	1.000	14	0	1020	1
Bagli Enrico	42	1.000	1.000	1.000	16	0	1019	0
Baldini Wander	196	0.998	0.989	0.989	89	1	944	1
Bandiera Laura	31	1.000	1.000	1.000	9	0	1026	0
Bettoni Diego	382	0.999	0.994	1.000	162	0	872	1
Calabrese Roberto	522	0.999	1.000	0.996	223	1	811	0
Cibinetto Gianluigi	370	1.000	1.000	1.000	157	0	878	0
Ciullo Giuseppe	96	0.997	0.923	1.000	36	0	996	3
Fioravanti Elisa	164	0.998	1.000	0.974	76	2	957	0
Fiorentini Gianni	113	1.000	1.000	1.000	49	0	986	0
Fiorini Massimiliano	251	0.996	1.000	0.965	109	4	922	0
Garzia Isabella	122	1.000	1.000	1.000	58	0	977	0
Guidi Vincenzo	237	0.998	1.000	0.981	101	2	932	0
Lamma Evelina	71	1.000	1.000	1.000	31	0	1004	0
Lenisa Paolo	138	1.000	1.000	1.000	59	0	976	0
Luppi Eleonora	513	0.997	0.991	0.995	218	1	814	2
Malagú Cesare	40	0.998	0.889	1.000	16	0	1017	2
Mantovani Fabio	36	0.996	0.733	1.000	11	0	1020	4
Mantovani Filippo	31	0.997	0.923	0.857	12	2	1020	1
Mantovani Franco	11	0.998	0.500	1.000	2	0	1031	2
Mazzolari Andrea	72	1.000	1.000	1.000	25	0	1010	0
Petrucci Ferruccio	46	1.000	1.000	1.000	20	0	1015	0
Ricci Barbara	28	0.998	0.941	0.941	16	1	1017	1
Riguzzi Fabrizio	76	0.997	1.000	0.914	32	3	1000	0
Schifano Fabio	65	1.000	1.000	1.000	27	0	1008	0
Tomassetti Luca	206	0.999	0.989	1.000	88	0	946	1
Tripiccione Raffaele	51	0.998	0.917	1.000	22	0	1011	2
Vincenzi Donato	41	1.000	1.000	1.000	10	0	1025	0

Table 3: Summary of the results for the data-set $\mathscr{D}$, namely the number of documents used for training, the metrics and the confusion matrix, for each of the 28 cases/authors of interest.

number of documents available for training and validation (only 15 in total) and simultaneously the presence of two homonyms (Filippo and Fabio) with a significantly larger number of documents.

In order to further asses its performances, the ANN has been trained and validated with an additional data-set $\mathscr{D}'$ (70% training, 30% validation). It includes $\mathscr{D}$ and additional documents authored by three authors of interest, homonyms of a very common italian name: Paolo Rossi. The validation set consists of 1125 documents.

Table 4 summarizes the results with $\mathscr{D}'$, showing the number of documents used for training, the metrics and the confusion matrix, for each of the 31 cases/authors of interest. The average Accuracy is 99.8% the average Recall is 93.0% and the average Precision is 98.1%; the network performances with the 28 authors in common between $\mathscr{D}$ and $\mathscr{D}'$ are consistent in the two runs. In addition, the network is able to disambiguate with a good level of confidence the three Paolo Rossi homonyms.

Author	Training	$\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$	$\frac{\text{TP}}{\text{TP+FN}}$	$\frac{\text{TP}}{\text{TP+FP}}$	True	False	True	False
	Documents	Accuracy	Recall	Precision	Positive	Positive	Negative	Negative
Alberti Marco	37	0.996	0.733	1.000	11	0	1110	4
Bagli Enrico	38	0.998	0.900	1.000	18	0	1105	2
Baldini Wander	197	0.997	0.978	0.989	87	1	1035	2
Bandiera Laura	25	0.998	0.867	1.000	13	0	1110	2
Bettoni Diego	382	0.997	0.988	0.994	161	1	961	2
Calabrese Roberto	533	1.000	1.000	1.000	212	0	913	0
Cibinetto Gianluigi	371	0.999	1.000	0.994	156	1	968	0
Ciullo Giuseppe	96	0.998	1.000	0.951	39	2	1084	0
Fioravanti Elisa	159	0.995	1.000	0.931	81	6	1038	0
Fiorentini Gianni	114	1.000	1.000	1.000	48	0	1077	0
Fiorini Massimiliano	253	0.998	0.991	0.991	106	1	1017	1
Garzia Isabella	118	0.997	0.952	1.000	59	0	1063	3
Guidi Vincenzo	237	0.996	0.970	0.990	98	1	1023	3
Lamma Evelina	74	1.000	1.000	1.000	28	0	1097	0
Lenisa Paolo	138	0.998	1.000	0.967	59	2	1064	0
Luppi Eleonora	521	1.000	1.000	1.000	212	0	913	0
Malagú Cesare	37	0.997	0.952	0.909	20	2	1102	1
Mantovani Fabio	36	0.999	0.933	1.000	14	0	1110	1
Mantovani Filippo	31	0.998	0.923	0.923	12	1	1111	1
Mantovani Franco	11	0.998	0.500	1.000	2	0	1121	2
Mazzolari Andrea	66	0.999	1.000	0.969	31	1	1093	0
Petrucci Ferruccio	45	1.000	1.000	1.000	21	0	1104	0
Ricci Barbara	35	0.999	0.900	1.000	9	0	1115	1
Riguzzi Fabrizio	76	0.997	1.000	0.914	32	3	1090	0
Rossi Paolo (pd)	76	0.995	0.812	1.000	26	0	1093	6
Rossi Paolo (pi)	84	0.998	1.000	0.947	36	2	1087	0
Rossi Paolo (uk)	51	0.996	0.818	1.000	18	0	1103	4
Schifano Fabio	65	0.998	0.926	1.000	25	0	1098	2
Tomassetti Luca	206	0.998	0.978	1.000	87	0	1036	2
Tripiccione Raffaele	53	0.996	0.864	0.950	19	1	1102	3
Vincenzi Donato	32	0.997	0.842	1.000	16	0	1106	3

Table 4: Summary of the results for an extended data-set $\mathscr{D}'$, namely the number of documents used for training, the metrics and the confusion matrix, for each of the 31 cases/authors of interest. Three homonyms named "Paolo Rossi" are included in the data-set.

5. Conclusions and future prospects

In this contribution we have faced the problem of disambiguation of authorship of scientific publications indexed by DL using a neural network approach. We have described in detail how to design and trains the neural network, and we have used as data-set records coming from Scopus DL. The results show that our approach is able to achieve an accuracy if approximately 99%, with a recall value – fraction of true-positive out of all actual positive in the range of $[0.5\ldots1.0]$. This high variance is actually related to the number of documents available for each author, and – as expected – is low for those having a small number of documents and high for those with an high number of publications.

Although the results presented refer to Italian authors only, the methodology can be applied in principle to any language because authorship recognition relies on co-writers sets. On the other hand, it is also true that for specific languages, like Indian, Chinese or Vietnamese, more issues related to homonyms may arise, badly impacting the accurancy of results.

For future works we plan to experiment with deep-learning networks, using two or more hidden layers, and different activation functions, and measure the corresponding impact on quality of results.

References

[1] Anderson A. Ferreira, et al. *A brief survey of automatic methods for author name disambiguation*. SIGMOD Rec. 41, 2, 15-26, 2012.

[2] I. Bhattacharya and L. Getoor. *Collective entity resolution in relational data*, ACM TKDD, 1(1), 2007.

[3] I. Bhattacharya and L. Getoor. *A latent dirichlet model for unsupervised entity resolution*, In SDM, 2006.

[4] V. C. Klaas. *Who's who in the world wide web: Approaches to name disambiguation*, Master's thesis, Diplomarbeit, LMU München, Informatik, 2007.

[5] J.M. Zurada, *Introduction to Artificial Neural System*, PWS, Boston (1992).

[6] J. Patterson and A. Gibson, *Deep Learning*, O'Really, Sebastopol (2017).

[7] S. Agatonovic-Kustrin and R. Beresford, *Journal of Pharmaceutical and Biomedical Analysis* 22 (5), p 717 (2000).

[8] Tommaso Sgarbanti. *Riconoscimento e disambiguazione degli autori di pubblicazioni scientifiche tramite reti neurali*, Bachelor's Thesis, Computer Science, University of Ferrara, 2017.