Deep learning based approach for entity resolution in databases

Nihel Kooli, Robin Allesiardo, and Erwan Pigneul

PagesJaunes - Solocal Group Rennes, France
{nkooli,rallesiardo,epigneul}@pagesjaunes.fr

Abstract. This paper proposes a Deep Neural Networks (DNN) based approach for entity resolution in databases. This approach is mainly based on a record linkage process which aims to detect records that refer to the same entity. First, record pairs are represented by their word embedding using an N-gram embedding based method. Then, they are classified into matching or unmatching pairs using a DNN model. Three DNN architectures: Multi-Layer Perceptron, Long Short Term Memory networks and Convolutional Neural Networks are investigated and compared for this purpose. The approach is experimented on two databases. The results exceed 97% for recall and 96% for precision. The comparison with similarity measure and classical classifier based approaches shows a significant improvement in the results on the two databases.

Keywords: Entity resolution, Databases, Record linkage, Deep Neural Networks, Word embedding, Similarity measures.

1 Introduction

A database is a repository that can merge records from several data sources with heterogeneous data formats. For example, the collection of a professional database from different telecommunication carriers or the collection of a scientific publication database from different archive websites. It may be also updated dynamically and/or managed by different users. This leads to duplicated, incomplete and erroneous data. Indeed, record attributes may be absent, may be non-normalized (abbreviations, acronyms, punctuation, etc.), may contain typographical errors or may have various entries (such as a professional which has two phone numbers).

To manage this data, such as using the database as a referential for a query-based search system, it is necessary to clean it. This is achieved by the Entity Resolution (ER) process which aims to synthesize records, to remove the redundancy and to identify the various entries of the same attribute. ER consists of detecting records that refer to the same entity, where an entity represents an existing or real thing, such as a person, a location, an organization, etc. Each entity is defined by a set of attributes. For instance, a person has the attributes: name, date of birth, social security number, etc.

The ER problem is also known in the databases community under the name of merge/purge or record linkage. It is often performed by the attribute comparison using similarity measures to tolerate their various representations.

Table 1. An extract of the publications database showing an example of two records referring to the same entity

Field	Record 1	Record 2
Title	Simple Greedy Matching for Aligning Large Knowledge Bases	SiGMa
Authors	Simon Lacoste-Julien; Konstantina Palla; Alex Davies; Gjergji Kasneci; Thore Graepel; Zoubin Ghahramani	S. Lacoste-Julien; K. Palla; A. Davies; G. Kasneci; T. Graepel; Z. Ghahramani
Affiliations	INRIA; University of Cambridge; University of Cambridge; Microsoft Research; Microsoft Research; University of Cambridge	(null)
Production date	2013-08-11	2013
Journal	(null)	(null)
Pages	pp. 572-580	572-580
Conference	The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining	KDD 2013
Conference date	2013-08-11	2013

Table 1 shows an extract of the scientific publication database used in our experiments. It represents two records referring to a same scientific publication. These records present several attribute dissimilarities, such as the acronyms in the title and the conference name, the representation of the author first names by their initials and the lack of some attributes (represented by "(null)").

String similarity measures [1], traditionally used for the ER purpose, are insufficient to overcome problems related to acronyms and abbreviations. Indeed, the edit distances, such as Levenshtein and Jaro-Winkler, are able to overcome elementary variations on characters. The bag of words distances, such as Jarccard and Tf-idf, are able to overcome term permutations. The hybrid distances, such as Monge-Elkan and Soft-tf-idf, are able to overcome term permutations with some character variations. But, none of these measures could detect that "The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining" and "KDD 2013" represent the same conference name. A system that is able to detect the similarity between these non-normalized representations of attributes is then required.

Recently, deep learning models have been successfully applied to the domains of text mining, natural language processing and computer vision [2]. These models have shown their power in learning features for several tasks. In this paper, we propose an ER approach based on word embedding for records representation and Deep Neural Networks (DNN) for record linkage learning and prediction.

The approach is experimented on two databases. The first database is in French language and represents contact information of professionals in our company's repository. The second one is in English language, represents meta-data of scientific publications and is extracted from public websites. Several deep neural networks are experimentally compared for this purpose. The comparison with three supervised learning classifiers (SVM, C4.5, Naives bayes) using similarity measure combination for attribute matching, shows the interest of employing a DNN model for the ER task.

The remainder of this paper is organized as follows. First, an overview of existent ER approaches is proposed in section 2. Second, our DNN ER based approach is detailed in Section 3. Then, experiments on two real world databases are presented in Section 4. Finally, section 5 concludes and suggests future works.

2 State of the art

A detailed literature review of ER approaches is provided in [3]. These approaches could be categorized into deterministic or probabilistic ones.

Deterministic approaches, such as that proposed in [4], are based on a set of rules fixed by experts which determinate the matching conditions of record pairs. These rules depend generally on a set of relevant fields. Similarity measures are generally employed in the comparison of attributes in order to tolerate the typographical errors. These techniques are time-consuming because they require significant human involvement. Furthermore, the predefined rules are very dependent on the database and on the domain.

Probabilistic approaches treat the problem as a classification one. This may be classifying records into entities or classifying record pairs into matching or unmatching ones. These approaches may be unsupervised or supervised.

Unsupervised approaches are useful when there is no annotated data. Most of them are inspired by the Fellegi and Sunter model described in [5]. It proposes to estimate matching and unmatching probabilities between attributes pairs based on a statistical study of the database. These probabilities are then used to calculate a matching score. The latter is compared to a threshold to make the matching decision.

Supervised approaches use annotated data to train a classification model. Such a method is described in [6] and employs two classification levels using an SVM. The first level represents attribute comparison using the Levenshtein distance, while the second level represents record comparison based on attribute similarity learning. The work in [1] combines 7 similarity measures to compute the similarity between attributes and uses a tree based classifier (C4.5) to match record pairs. Authors in [7] use active learning based techniques to select the most informative record pairs. Users are solicited to annotate these pairs as matching or unmatching ones in order to train the classifier. The latter generates classification rules that intersect attributes and their similarity measures.

To our knowledge, there is no ER method that uses deep learning for the complete process of record linkage. Authors in [8] propose a hybrid human/machine

approach. It takes as input a record and proposes to match it with its corresponding entity. First, candidate entities are selected for each record based on a single layered convolutional neural network. The input of this network is a vector representing the word embedding of its relevant words. Then, candidate entities are analyzed by human experts based on a crowd-sourcing approach. Even if the cost of the crowdsourcing is reduced by the deep learning network, this approach still requires human intervention. Furthermore, word embedding is realized using word2vec [9] which can not output a vector for a word that is not in the pre-trained model and does not take into account word syntactic variation. This approach was experimented on only 300 records.

In the domain of name disambiguation, the work in [10] proposes an approach for integrating Vietnamese author names in different publications. It uses a multilayer perceptron which takes as input record pairs and proposes to classify it into a matching or an unmatching ones. Each record pair is represented by a vector of similarity distances between attribute pairs. The problem with this method is that is dependent on the employed similarity measures. This approach was experimented on about 4300 records.

Our novel approach is completely automatic and does not require any human intervention. In addition, the attribute comparison is independent of any similarity measure choice and gets over their deficiency in particular cases. This similarity is automatically learned by the DNN.

3 Deep learning based entity resolution approach

Our ER approach is performed by comparing record pairs in order to link those referring to the same entity. To reduce the number of comparisons, a preliminary phase called database segmentation is integrated.

In the following, we detail the database segmentation process employed in this approach. Then, we present a novel record linkage method based on DNN. Finally, we explain the generation process of the training dataset.

3.1 Database segmentation

In large databases, record pairs comparison is expensive since it is a Cartesian product of database records. Database segmentation [11] is often used to solve this problem. It consists of grouping into blocks close records that are likely to represent the same entity. Therefore, only the records of a same block are compared two by two in the record linkage step. This grouping is performed using grouping keys. A grouping key may be a given field or a combination of given fields. The records whose attributes corresponding to these keys are similar will be grouped together. The n first characters of a company's "name" field combined with the zip code in professional database and the n first terms of the "title" field of a publication in scientific publication database are good examples of such keys.

3.2 Record linkage

The global model of the record linkage approach is represented in Fig. 1. This model takes as input pairs of records and proposes to classify them into "matching" or "unmatching" ones. Firstly, a pre-processing step is performed. It consists of extracting a vector of key words and representing it by a numerical matrix using a word embedding step. Secondly, the deep neural network is trained. The first layers of this network consists of learning the classification features while the last one corresponds to the binary classification process. Record linkage steps will be explained in the following.

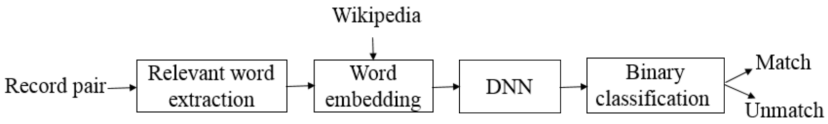

Fig. 1. Entity resolution approach model

Relevant word extraction represents a pre-processing step of the record attribute data. It consists of removing empty words, such as "the", "and", "of", etc. Indeed, these words do not contribute to the training of the DNN since they are not relevant in the process of record matching and may degrade the efficiency of the network. In addition, we propose to remove punctuation and special characters.

Word embedding is used to represent textual data in the attributes by dense real valued vectors with a notion of distance between words. One common way to achieve such embedding is to use the word2vec approach [9]. This approach is very efficient but suffer from a major drawback in our setting, as only words from the dictionary can be embedded. However, in the ER task, data is often non-normalized and many records can contain new words not previously seen in the dataset used to learn the embedding.

To overcome this issue, we instead use a N-gram based embedding method, available through the Fasttext library [12]. Fasttext is similar to word2vec but adds information about subwords (the N-grams). This additional information is available for every words, yielding to a more robust representation of new words.

In this work, we trained the Fasttext word embedding model on the wikipedia data ("wiki.fr" for the French database and "wiki.en" for the English database). The word embedding size is empirically fixed to 100.

Deep neural network is used to decide the matching of two records. Several architectures can be used to achieve this goal. We discuss some of them in this section and compare their performances later in the experiments. We generically call them DNN and consider their type as a parameter of the algorithm. Regardless of their type, every architecture takes as input a record pair $r_i r_j$ and exposes a binary classifier as their output layer.

We now formalize the data processing in input of the DNN. A record is compounded of several attributes, themselves compounded of several words. Let n be the maximum number of words by attribute; words in excess are truncated. For each attribute, the word embeddings are concatenated into a real valued vector of size $D \times n$, where D is the size of the word embedding. When the number of words is lesser than n, the end of the vector is padded with zeros. The record pair $r_i r_j$ of two entities is the matrix obtained by the concatenation of the attribute vectors of both records (one row is an attribute vector). We detail the three used DNN architectures in the following.

The Fully Connected Network: is a regular Multi Layer Perceptron (MLP) [13] where each neuron of each layer is fully connected to the neurons of the previous and the next layer. For the record pairs classification, we use an MLP composed of k dense layers: the input layer, $k-2$ hidden layers and the output layer. k is empirically fixed to 4 using cross-validation. The number of units in the input layer is $2 \times D \times n \times m$, where m is the number of database fields. The number of units in the hidden layers is empirically fixed to 100. The input and hidden layers are followed each by a Batch Normalization layer to normalize the next layer inputs. The activation layer is carried out using the rectifier function $f(x) = \max(x, 0)$ for the input and the hidden layers. Each activation layer is followed by a Dropout using a rate of 0.5. The output layer is composed of two units which corresponds to the cases of "matching" or "unmatching" record pairs. The activation layer is carried out using the sigmoid function $f(x) = sigmoid(x) = \frac{1}{1+e^{-x}}$. The gradient descent is performed by a Rooted Mean Square (RMS) back-propagation for 100 epochs with mini batches of size 100.

Long Short Term Memory Networks (LSTM): [15] were created to allow the network to maintain a memory of the previous inputs. Whereas a regular DNN only uses the current input for the prediction, the LSTM can process the input as a sequence. To our knowledge, LSTM models has never been used for the task of record linkage. In this work, we sequentially feed the LSTM with the embedding of the attributes in the record pair. The number of sequences is then equal to $2 \times m$. The used LSTM network contains 3 LSTM hidden layers with 32 hidden units. It has been trained with the Adam optimizer [16] for 70 epochs with a batch size of 200.

Convolutional Neural Networks (CNN): were popularized due to their performances on image classification [17]. The convolutional layers allow to train classifier on image with minimal pre-processing. Filters convolving on the input are being learned by the network and replace hand-engineered features. Recently, convolutional networks were successfully used on textual data [14]. In this work, we use 4 1D convolutional layers followed by a fully connected layer for the record pair classification (see Fig. 2). Each convolutional layer uses a filter of size 4 and a stride of 1. These parameters are empirically tuned. The first three convolutional layers are followed each by a max pooling layer while the last convolutional layer is followed by an average pooling layer. The CNN is regularized

using a Dropout with a rate of 0.7. The activation layer is carried out using the ReLU function for the all layers except the last one where a sigmoid is used. The gradient descent is performed by Stochastic Gradient Descent (SGD) for 100 epochs with mini-batches of size 200.

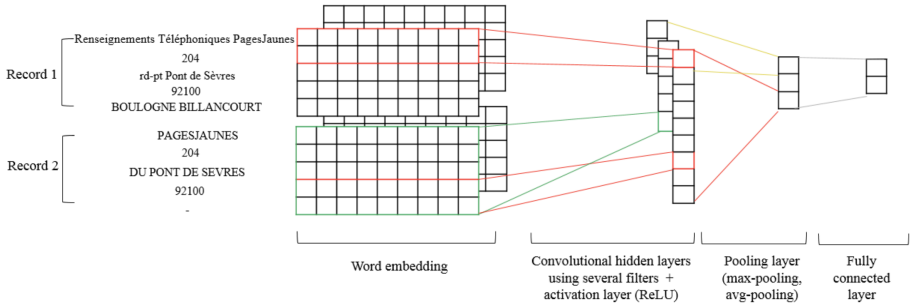

Fig. 2. Convolutional neural network model for record pair classification

3.3 Dataset generation

The training dataset is generated based on coupling the records pairs of the same database block. Since there is much more non-similar record pairs than similar ones, we obtain a dataset with imbalanced classes distribution. To resolve this problem, we use cluster-based over sampling. This consists of clustering matching and unmatching classes independently using the k-means algorithm. Thus, record pairs samples of each cluster are duplicated such that all clusters of each class have the same number of samples and the two classes have the same number of samples. The cluster-based over sampling performed better than simple over-sampling and under-sampling methods in our experiments.

4 Experiments

4.1 Datasets

Professional database contains entities that represent professionals in our company's repository such as societies, doctors, restaurants, plumbers, etc. It is collected from telecommunication carrier databases merged with the siren public database[1]. This database is in French language. Each professional is described by the attributes: denomination, street type, street number, street, zip code, city, region, geo-localization, siret number, etc. This database is composed of 11 478 587 records and 6 897 305 entities. The dataset used for the experiments contains 17 975 367 records, decomposed as shown in Table 2.

[1] https://www.sirene.fr/

Scientific publication database contains meta-data of scientific publications such as journals, conferences, thesis manuscripts, posters, etc. It is extracted from public archive websites of scientific articles: HAL[2], ISTEX[3] and DBLP[4]. This database is in English language. The entity attributes are: title, authors, affiliations, pages, editors, conference, journal, dates, etc. The publication database is composed of 415 500 records and 286 695 entities. The datasets used for the experiments is composed of 930 800 as shown in Table 2.

Table 2. Datasets used in the experiments

Dataset	Professionals database	Publications database
Training (#records pairs)	10 785 220	558 000
Validation (#records pairs)	1 797 537	93 800
Test (#records pairs)	5 392 610	279 000
Total (#records pairs)	17 975 367	930 800

4.2 Record linkage results

Evaluation metrics: a matching pair is defined as a pair that refers the same entity in reality. A linked pair is defined as a record pair that is matched with our system. Recall, Precision are then defined in (1).

$$Recall = \frac{\#correctly\ linked\ pairs}{\#matching\ pairs}; \quad Precision = \frac{\#correctly\ linked\ pairs}{\#linked\ pairs} \tag{1}$$

Record linkage results: are reported in Table 3 for the two experimented databases. Three DNN architectures are compared for this purpose. These results show that the LSTM outperforms the two other DNN architectures on the professional database and that the CNN is the best architecture on the scientific publication database. False positives are essentially caused by different professionals having the same address. False negative are essentially caused by missing attributes in the records or the over segmentation of the database (placing similar records in different blocks discarding the possibility of comparing them).

To show the interest of employing the word embedding process, we replaced the word embedding resulting matrix by a one that contains the similarity measure values between the attributes pairs. Seven distinct similarity distances: Levenshtein, Jaro, Jaro-Winkler, Jaccard, Tf-idf, Monge-Elkan, Soft-tf-idf are used for this purpose (a study of these measures has been presented in [1]). The comparison results, reported in Table 3, show that the embedding process leads to a better performance. Indeed, learning a projection of the data to a new vector space is equivalent to learning a new similarity measure [18]. This depict the advantage of a learned similarity measure versus an arbitrary defined one.

[2] https://hal.archives-ouvertes.fr/
[3] http://www.istex.fr/
[4] http://dblp.uni-trier.de/

Table 3. Record linkage results using DNN

Input	Model	Professionals database			Publications database		
		Recall	Precision	F-measure	Recall	Precision	F-measure
Word embedding	MLP	98.00	98.00	98.00	95.60	96.98	96.29
	LSTM	**99.05**	**98.70**	**98.87**	96.76	95.89	96.32
	CNN	98.10	97.70	97.80	**97.45**	**96.78**	**97.11**
Similarity measures	MLP	91.74	98.80	94.89	93.51	91.97	92.73
	LSTM	91.18	98.70	94.50	94.56	93.76	94.16
	CNN	85.41	97.50	90.18	94.34	94.20	94.27

Comparison: our DNN based approach is compared with classical classifiers based approaches: SVM, C4.5 and Naive Bayes (as proposed in [1]). The used features for these classifiers are the 7 similarity measures presented above. The results are reported in Table 4. The comparison with Table 3 shows that our DNN approach significantly outperforms the classical classifiers and confirms its ability to learn the similarity between non-normalized entity attributes.

Table 4. Record linkage results using classical classifiers [1]

Similarity measures	Classifier	Professionals database			Publications database		
		Recall	Precision	F-Measure	Recall	Precision	F-Measure
Monge-Elkan	SVM	88.60	90.30	89.44	93.01	92.90	92.95
	C4.5	89.80	88.01	88.89	93.30	92.95	93.12
	Naive Bayes	91.25	89.45	90.34	91.00	89.96	90.48
All measures	SVM	89.57	91.76	90.65	93.40	93.00	93.20
	C4.5	91.25	92.45	91.85	94.90	94.56	94.73
	Naive Bayes	92.78	90.54	91.64	92.98	92.20	92.59
Monge-Elkan +	SVM	89.57	91, 76	90.65	93.40	93.00	93.20
Levenshtein +	C4.5	**92.00**	**92.00**	**92.00**	**94.90**	**94.56**	**94.73**
Jaro-Winkler + Tf-idf	Naive Bayes	92.78	90.54	91.64	92.98	92.20	92.59

5 Conclusion

In this paper, we proposed an ER approach based on word embedding and DNN models. The use of N-gram embedding and the automatic learn of attribute similarity by the DNN has proven to be effective in improving the record linkage process. The results on two entity databases (the first one is in the French language where the second one is in the English language) are promising and exceed 97% for recall and 96% for precision.

Our future work is to evaluate our approach on other public databases in order to investigate more attribute variability. In addition, we plan to deal with relational databases, where the entities are described by multiple tables connected by foreign keys. Another perspective is to extend the system to handle with the incremental update of databases.

References

1. Kooli, N.: Data matching for entity recognition in ocred documents. Thesis defense, Lorraine university (2016)
2. Schmidhuber, J.: Deep learning in neural networks: An overview. In: Neural Networks, vol 61, pp. 85-117 (2015)
3. Christen, P.: Data matching - concepts and techniques for record linkage, entity resolution, and duplicate detection. In: Data-Centric Systems and Applications Description, pp. 1–270 (2012)
4. Lee, M.L. , Ling, T.W. , Low, W.L.: Intelliclean: A knowledge-based intelligent data cleaner. In: Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining, pp. 290–294 (2000)
5. Fellegi, I., Sunter, A.: A theory for record linkage. In: Journal of the American Statistical Association, vol. 64, pp. 1183–1210 (1969)
6. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 39-48 (2003)
7. Tejada, S., Knoblock, C. A., Minton, S.: Learning domain-independent string transformation weights for high accuracy object identification. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 350–359 (2002)
8. Gottapua, R.D., Daglia, C., Ali, B.: Entity Resolution Using Convolutional Neural Network. In: Procedia Computer Science, vol. 95, pp. 153–158. Elsevier (2016)
9. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. At: http://arxiv.org/abs/1301.3781 (2013)
10. Huynh, T. , Hoang, K., Do, T., Huynh, D.: Author Name Disambiguation by Using Deep Neural Network. In: Asian Conference on Intelligent Information and Database Systems, vol. In: 7802 of Lecture Notes in Computer Science, pp. 226–235. Springer, Kuala Lumpur (2001)
11. Bilenko, M.: Adaptive blocking: Learning to scale up record linkage. In: Proceedings of the 6th IEEE International Conference on Data Mining, pp. 87–96 (2006)
12. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword. In: Transactions of the Association for Computational Linguistics, vol. 5, pp. 135-146 (2017)
13. Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain". In: Psychological Review. vol. 65 (6): 386–408.
14. Collobert, R.: Deep Learning for Efficient Discriminative Parsing. In: 21st International Conference on Artificial Intelligence and Statistics, pp. 224-232 (2011)
15. Hochreiter, S., Schmidhuber, J.: Long short-term memory. In: Neural computation (1997)
16. Kingma, D.P., Ba, J.: Distributed Representations for Biological Sequence Analysis. In: Data and Text Mining in Biomedical informatics, abs/1412.6980 (2016)
17. Krizhevsky, A., Sutskever, I., Geoffrey E. Hinton: ImageNet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems 25 - NIPS (2012)
18. Yih, W., Meek, C.: Learning Vector Representations for Similarity Measures. In: Microsoft Technical Report MSR-TR-2010-139 (2010)