Who wrote this book? A challenge for e-commerce

Béranger Dumont, Simona Maggio, Ghiles Sidi Said & Quoc-Tien Au
Rakuten Institute of Technology Paris
{beranger.dumont,simona.maggio}@rakuten.com,
{ts-ghiles.sidisaid,quoctien.au}@rakuten.com

Abstract

Modern e-commerce catalogs contain millions of references, associated with textual and visual information that is of paramount importance for the products to be found via search or browsing. Of particular significance is the book category, where the author name(s) field poses a significant challenge. Indeed, books written by a given author might be listed with different authors' names due to abbreviations, spelling variants and mistakes, among others. To solve this problem at scale, we design a composite system involving open data sources for books, as well as deep learning components, such as approximate match with Siamese networks and name correction with sequence-to-sequence networks. We evaluate this approach on product data from the e-commerce website Rakuten France, and find that the top proposal of the system is the normalized author name with 72% accuracy.

1 Introduction

Unlike brick-and-mortar stores, e-commerce websites can list hundreds of millions of products, with thousands of new products entering their catalogs every day. The availability and the reliability of the information on the products, or *product data*, is crucial for the products to be found by the users via textual or visual search, or using faceted navigation.

Books constitute a prominent part of many large e-commerce catalogs. Relevant book properties include: title, author(s), format, edition, and publication date, among others. In this work, we focus on the names of book authors, as they are found to be extremely relevant to the user and are commonly used in search queries on e-commerce websites, but suffer from considerable variability and noise. To the best of our knowledge, there is no large-scale public dataset for books that captures the variability arising on e-commerce marketplaces from user-generated input. Thus, in this work we use product data from Rakuten France (RFR).[1]

The variability and noise is evident in the RFR dataset. For example, books written by F. Scott Fitzgerald are also listed with the following author's names: "Francis Scott Fitzgerald" (full name), "Fitzgerald, F. Scott" (inversion of the first and last name), "Fitzgerald" (last name only), "F. Scott Fitgerald" (misspelling of the last name), "F SCOTT FITZGERALD" (capitalization and different typological conventions), as well as several combinations of those variations. The variability of the possible spellings for an author's name is very hard to capture using rules, even more so for names which are not primarily written in latin alphabet (such as arabic or asian names), for names containing titles (such as "Dr." or "Pr."), and for pen names which may not follow the usual conventions. This motivated us to explore automated techniques for normalizing the authors' names to their best known ("canonical") spellings.

Fortunately, a wealth of open databases exist for books, making it possible to match a significant fraction of the books listed in e-commerce catalogs. While not always clean and unambiguous, this information is extremely valuable and enables us to build datasets of name variants, used to train machine learning systems to normalize authors' names. To this end, in addition to the match with open databases, we will explore two different approaches: approximate match with known authors' names using Siamese neural networks, and direct correction of the provided author's name using sequence-to-sequence learning with neural networks. Then, an additional machine learning component is used to rank the results.

The rest of the paper is organized as follows: we present the data from RFR and from the open databases in Section 2, before turning to the experimental setup for the overall system and for each of its components in Section 3. Finally, we give results in Section 4, we present related works in Section 5, and conclude in Section 6.

2 Book data

2.1 Rakuten France data

The RFR dataset contains 12 million book references[2]. The most relevant product data for normalization is:

[1] https://fr.shopping.rakuten.com

[2] The RFR dataset is publicly available at https://rit.rakuten.co.jp/data_release.

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 121–125
Hong Kong, Nov 4, 2019.

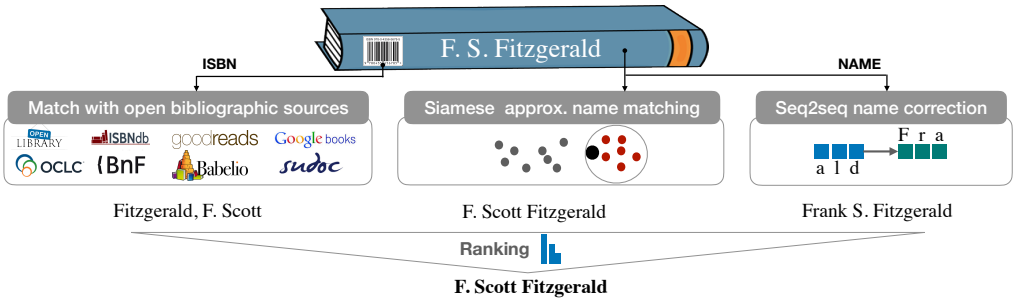

Figure 1: Overview of the system for normalizing author names. Each component is detailed in Section 3.

Table 1: Performances of the external bibliographic resources used for matching books on RFR via ISBN.

Source	URL	% of ISBNs
Open Library	openlibrary.org	24.9%
ISBNdb	isbndb.com	36.3%
Goodreads	www.goodreads.com	64.7%
Google Books	books.google.com	51.2%
OCLC	www.oclc.org	52.2%
BnF	www.bnf.fr	7.4%
Sudoc	www.sudoc.abes.fr	29.0%
Babelio	www.babelio.com	7.9%

- **ISBN**[3] in 10 digit or 13 digit format;
- **product title**, which includes the book title, often supplemented with extra information in free text;
- **author(s)** of the book as the input catalog name provided by the seller.

In particular, the ISBN is a worldwide unique identifier for books, which makes it a prime candidate for unambiguous matching with external sources. In this dataset, an ISBN is present for about 70% of the books. Among the books with no ISBN, 30% are ancient books which are not expected to be associated an ISBN.

2.2 External bibliographic resources

There is no central authority providing consistent information on books associated with an ISBN. However, there is a wealth of bibliographic resources and open databases for books. In order to retrieve the author's name(s) associated with the books in the RFR dataset, we perform ISBN matching using public APIs on eight of them, listed in Table 1 along with the fraction of found ISBNs from this dataset. We find the sources to be highly complementary and that 75% of the books with an ISBN are matched with at least one source. The match via ISBN on external bibliographic resources is the first component of the system depicted in Fig. 1.

2.3 Dataset of name entities

In order to train and evaluate machine learning systems to match or correct authors' names, a dataset of name entities containing the different surface forms (or variants) of authors' names is required. The entities should reflect as well as possible the variability that can be found in the RFR dataset, as was illustrated in the case of F. Scott Fitzgerald in Section 1.

For each entity, a canonical name should be elected and correspond to the name that should be preferred for the purpose of e-commerce. Instead of setting these gold spellings by following some predefined rules (i.e. family name in the first position, initial of first name, etc.), for e-commerce applications it is more appropriate that the displayed authors names have the most popular spellings among readers. In agreement with Rakuten catalog analysts we set the most popular spelling of an author name as the one found on Wikipedia[4] or DBpedia (Lehmann et al., 2015).

While Wikipedia seems more pertinent to select canonical names matching the e-commerce user expectations, specialized librarian data services, such as the Library of Congress Name Authority[5], could be used in future research to enrich the dataset of name entities.

Name entities are collected in three distinct ways:

1. **ISBN matching**: for each book the different author names found via ISBN search on external sources and the RFR author name field build up an entity. The canonical form is the one that is matched with Wikipedia or DBpedia; else the one provided by the greatest number of sources.

2. **Matching of Rakuten authors**: we build entities using fuzzy search on the author name field on DBpedia and consider the DBpedia value to be canonical. We limit the number of false positives in fuzzy search by tokenizing both names, and keeping only the names where at least one token from the name on RFR is approximately found in the external resource (Levenshtein distance < 2).

3. **Name variants**: DBpedia, BnF, and JRC-names (Steinberger et al., 2011; Maud et al., 2016) directly provide data about people (not limited to book authors) and their name variants.

[3] International Standard Book Number, see https://www.isbn-international.org

[4] https://www.wikipedia.org

[5] id.loc.gov/authorities/names.html

As an example, by using the wikiPageRedirects field in DBpedia we can build a large entity for the canonical name "Anton Tchekhov", containing "Anton Tchechov", "Antòn Pàvlovič Chéchov", "Checkhov", "Anton Chekov", and many more.

After creating the name entity dataset, we normalize all names to latin-1. We obtain about 750,000 entities, for a total of 2.1 million names.

2.4 Annotated Rakuten France data

In order to evaluate the overall system, we need product data from RFR for which the canonical author name has been carefully annotated and can be considered as the ground truth. To this end, we have considered a subset of 1000 books from the RFR dataset, discarding books written by more than one author for simplicity.[6] We find that 467 books have a canonical author name that differs from RFR's original (unnormalized) author name. Also, 310 do not have an ISBN or do not match on any of the bibliographic resources listed in Section 2.2. Among them, 208 books have a canonical name that differs from the input catalog name provided by the seller.

3 Experimental setup

The overview of the system can be found in Fig. 1. Its first component, the matching via ISBN against external databases, has already been presented in Section 2.2. In the rest of this section, we will shed light on the three machine learning components of the system.

3.1 Siamese approximate name matching

We want to learn a mapping that assigns a similarity score to a pair of author names such that name variants of the same entity will have high similarity, and names that belong to different entities will have low similarity. Once learned, this mapping will enable us to assign an entity to any given name.

To this end, we might use a classical string metric such as the Levenshtein distance or the n-gram distance (Kondrak, 2005). However, those are not specific to people's names, and might return a large distance (low similarity) in cases such as the inversion between first name and last name or the abbreviation of the first name to an initial. Thus, we want to use the dataset of name entities to learn a specialized notion of similarity—this is known as distance metric learning (Kulis et al., 2013).

To this purpose, we use a pair of neural networks with shared weights, or Siamese neural network (Bromley et al., 1994). Each network is a recurrent neural network (RNN) composed of a character-level embedding layer with 256 units, a bidirectional long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) with 2×128 units, and a dense layer with 256 units. Each network takes a name as input and outputs a representation—the two representations are then compared using cosine similarity with a target value equal to 1 for name variants of the same entity, and to 0 otherwise. We preprocess the input by representing all characters in ASCII and lowercase. We consider a sequence length of 32 using zero padding.

The Siamese network is trained with contrastive loss (Hadsell et al., 2006) in order to push the similarity towards 1 for similar pairs, and below a certain margin (that we set to 0) for dissimilar pairs. The optimization is done using Adam (Kingma and Ba, 2014), with a learning rate of 10^{-3} and a gradient clipping value of 5. We use batches of 512 samples, consider a negative to positive pairs ratio of $4:1$, and randomly generate new negative pairs at every epoch.

At test time, we search for the canonical name whose representation is closest to that of the query, using only the high-quality name entities from DBpedia, BnF, and JRC-names. To this end, we do approximate nearest neighbor search using Annoy[7].

3.2 Name correction with seq2seq networks

We use a generative model to correct and normalize authors' names directly. The dataset of name entities is again employed to train a sequence-to-sequence (seq2seq) model (Sutskever et al., 2014) to produce the canonical form of a name from one of its variants. The dataset is further augmented by including additional variants where the first name is abbreviated to an initial.

The seq2seq model is an encoder-decoder using RNNs, with a character embedding layer, as in the case of the Siamese network. The encoder is a bi-directional LSTM with 2×256 units, while the decoder is a plain LSTM with 512 units connected to a softmax layer that computes a probability distribution over the characters.

The training is performed by minimizing the categorical cross-entropy loss, using teacher forcing (Williams and Zipser, 1989). The optimization setting is identical to that of the Siamese nework, with batches of 1024 samples. For inference, we collect the 10 output sequences with highest probability using beam search.

3.3 Ranking of the proposals

For any given book with an ISBN and an author's name, all three techniques shown in Fig. 1 provide one or several candidate canonical names. As we aim at providing an automated tool to enhance the quality of the book products, the final system should provide a ranked list of candidates with a calibrated confidence level. For this purpose we train a logistic regression to estimate the probability that a proposal is the canonical form for an author's name. This information is then used as a confidence score to rank the different candidate names returned by the three normalization approaches.

Specifically, we represent a proposal with a set of 12 features: 11 indicating whether it is found in the bibliographic sources, generated from the seq2seq model, matched with the Siamese network or equal to the input name, and one last feature corresponding to the cosine

[6] The annotated RFR dataset is publicly available at https://rit.rakuten.co.jp/data_release.

[7] https://github.com/spotify/annoy

distance between the representation of the proposal and that of the input name. The selected features reflect that the confidence of the global system should increase with *(i)* the consensus among the different sources, and *(ii)* the similarity of the candidate to the input name.

For this component we use the annotated dataset introduced in Section 2.4, splitting the books between training and test sets, with a ratio of $50\% : 50\%$, generating a total of 11185 proposals.

4 Results

The three machine learning components discussed in the previous section have been individually evaluated on their specific task. Furthermore the final system has been evaluated in terms of correctly normalized book authors in a real case scenario.

Siamese approximate name matching We evaluate the Siamese network on a held out test set, and compare it to an n-gram distance, by checking that the nearest neighbor of a name variant is the canonical name of the entity to which it belongs. We find an accuracy of 79.8% for the Siamese network, against 71.1% for the n-gram baseline with $n = 3$. We have also checked metrics when introducing a threshold distance above which we consider that no matching entity is found, and found systematic improvement over the baseline. In the final system, we set the threshold to infinity.

Siamese networks are more effective than simpler rule-based approaches and more specifically they perform better than the n-gram baseline on the following cases:

- Vittorio Hugo → Victor Hugo: capturing name variants in different languages;
- Bill Shakespeare → William Shakespeare: capturing common nicknames

Name correction with seq2seq networks Similarly to the previous approach, the seq2seq network is evaluated on a held out test set by checking that one of the generated name variants is the canonical name of the entity to which it belongs. As expected, name normalization using seq2seq network gives poorer performances than approximate matching within a dataset of known authors, but constitutes a complementary approach that is useful in case of formatting issues or incomplete names. This approach alone reaches a top-10 accuracy of 42% on the entire test set, 26% on a test set containing only names with initials, and 53% on a test set containing only minor spelling mistakes.

Some examples where seq2seq performs better than the other methods are as follows:

- V. Hugo → Victor Hugo: first name prediction for authors we don't have in the canonical database;
- Vicor Hugo → Victor Hugo: misspelling correction for authors we don't have in the canonical database.

Table 2: Global system top-k accuracy at the book level.

Type of books	#samples	acc@1	acc@3
all	500	72%	85%
unnorm. input author	235	49%	67%
no ISBN match	151	50%	64%
unnorm. + no ISBN	109	35%	49%

Ranking of the proposals With a decision threshold of $p = 0.5$, the trained classifier has an accuracy of 93% for both positive and negative candidates in the test set. The coefficients of the estimator reveal the importance of the features and, thus, of the related components. The three most important contributions are the match with the Siamese network, the match via ISBN in Babelio, and the similarity with the input catalog name, confirming the relevance of a multi-approach design choice.

Global system In order to reflect the actual use of the global system on e-commerce catalog data, the final evaluation is performed at the book level, by considering all the proposals provided by the different components for a given book. The metric used is the top-k accuracy on the ranked list of proposals for each book; results are summarized in Table 2. We find that 72% of the books have the author's name normalized by the highest ranked proposal. Excluding from the evaluation books where the ground truth for the author's name equals the catalog value, this accuracy drops to 49%. In the case of books without ISBN or that do not match on any of the bibliographic resources, thus relying on machine learning-based components only, we find that 50% of the books are normalized by the top proposal. Finally, for the combination of the above two restrictions, we find a top-1 accuracy of 35%.

5 Related works

There is a long line of work on author name disambiguation for the case of bibliographic citation records (Hussain and Asghar, 2017). While related, this problem differs from the one of book authors. Indeed, unlike most books, research publications usually have several authors, each of them having published papers with other researchers. The relationships among authors, which can be represented as a graph, may be used to help disambiguate the bibliographic citations.

Named entity linking (Shen et al., 2015), where one aims at determining the identity of entities (such as a person's name) mentioned in text, is another related problem. The crucial difference with the disambiguation of book authors is that entity linking systems leverage the context of the named entity mention to link unambiguously to an entity in a pre-populated knowledge base.

The conformity of truth in web resources is also a related problem, addressed in the literature by TruthFinder (Yin et al., 2008) algorithms. Similarly, the proposed global model in which we combine sources learns to some extent the level of trust of the different

sources. Unlike our technique, the TruthFinder approach needs to start from a book we can unambiguously identify in several sources and, thus, needs its ISBN.

Distance metric learning with neural networks has been used for merging datasets on names (Srinivas et al., 2018), for normalizing job titles (Neculoiu et al., 2016), and for the disambiguation of researchers (Zhang et al., 2018). Sequence-to-sequence learning has been used for the more general task of text normalization (Sproat and Jaitly, 2016), and for sentence-level grammar error identification (Schmaltz et al., 2016).

To the best of our knowledge, the problem of normalization of book authors name has not been tackled in the previous literature, except for a work on named entity linking for French writers (Frontini et al., 2015).

6 Conclusions

We provided a first attempt at solving the problem of author name normalization in the context of books sold on e-commerce websites. To this end, we used a composite system involving open data sources for books, approximate match with Siamese networks, name correction with sequence-to-sequence networks, and ranking of the proposals. We find that 72% of the books have the author's name normalized by the highest ranked proposal.

In order to facilitate future research, we are releasing data from Rakuten France: a large dataset containing product information, and a subset of it with expert human annotation for the authors' names. They are accessible at `rit.rakuten.co.jp/data_release`.

Multiple challenges remain and are left for future research. First, the system should be extended to handle the case of books with multiple authors. In addition, the book title could be used to help disambiguate between authors and to query external bibliographic resources. This work can also be seen as an intermediate step towards a knowledge base for book author names with name variants, extending public ones such as BnF, using the ISNI[8] for easier record linkage whenever available.

Acknowledgments

We thank Raphaël Ligier-Tirilly for his help with the deployment of the system as microservices, and Laurent Ach for his support.

References

J. Bromley et al. 1994. Signature verification using a "siamese" time delay neural network. pages 737–744.

F. Frontini et al. 2015. Semantic web based named entity linking for digital humanities and heritage texts.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping.

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

I. Hussain and S. Asghar. 2017. A survey of author name disambiguation techniques: 2010 to 2016. *The Knowledge Engineering Review* 32.

D. P. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization .

G. Kondrak. 2005. N-gram similarity and distance. Springer, pages 115–126.

B. Kulis et al. 2013. Metric learning: A survey. *Foundations and Trends in Machine Learning* 5(4):287–364.

J. Lehmann et al. 2015. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal* 6(2):167–195.

E. Maud et al. 2016. Jrc-names: Multilingual entity name variants and titles as linked data. *Semantic Web Journal* 8(2):283–295.

P. Neculoiu et al. 2016. Learning text similarity with siamese recurrent networks.

A. Schmaltz et al. 2016. Sentence-level grammatical error identification as sequence-to-sequence correction.

W. Shen, J. Wang, and J. Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE TKDE* .

R. Sproat and N. Jaitly. 2016. Rnn approaches to text normalization: A challenge .

K. Srinivas, A. Gale, and J. Dolby. 2018. Merging datasets through deep learning .

R. Steinberger et al. 2011. Jrc-names: A freely available, highly multilingual named entity resource.

I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks.

R. J. Williams and D. Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1(2):270–280.

X. Yin et al. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering* .

Y. Zhang et al. 2018. Name disambiguation in aminer: Clustering, maintenance, and human in the loop.

[8] International Standard Name Identifier, `isni.org`