Rubim and Braganholo *Journal of the Brazilian Computer Society* (2017) 23:3
DOI 10.1186/s13173-017-0052-0

Journal of the
Brazilian Computer Society

RESEARCH

Open Access

Detecting referential inconsistencies in electronic CV datasets

Ivison C. Rubim[1] and Vanessa Braganholo[2*]

Abstract

One way to measure the scientific progress of a country is to evaluate the curriculum vitae (CV) of its researchers. In Brazil, this is not different. The Lattes Platform is an information system whose primary objective is to provide a single repository to store the CV of the Brazilian researchers. This system is increasingly acquiring expressiveness as the main source of information regarding the Brazilian community of researchers, students, managers, and other actors in the national system of science, technology, and innovation. However, the integrity of this important tool for gaging the national bibliographic production may be affected by the effect of ambiguities or referential inconsistencies in coauthoring citations. A first step towards solving this problem lies in identifying such inconsistencies. For that, we propose a heuristic-based approach that uses similarity search to match papers from coauthors of CV. We then use this technique to analyze over 2000 curricula of researchers from a given institution recovered from the Lattes Platform. The results indicate 18.98% of the analyzed publications present referential inconsistencies, which is a significant amount for a dataset that is supposed to be correct and trustable.

Keywords: Electronic curricula, Lattes, Inconsistency, Similarity

Introduction

Researchers are, and always will be, a valuable resource of wealth for any country. They are the main responsible for innovation and scientific discoveries that can change the course of the world and have a significant impact on the quality of human life. In that matter, one way to measure the scientific progress of a country is to evaluate the curriculum vitae (CV) of its researchers.

In Brazil, this is not different. In fact, in the mid-80s, the Brazilian National Council for Scientific and Technological Development (CNPq) started to design a database of electronic CV. Researchers could enter data into this database using a standard electronic form. At that moment, governmental agencies were eager for this kind of data, arguing that it could aid the definition of national policies for supporting science and technology. This effort evolved into an information system named Lattes Platform.[1] Lattes Platform is rich in terms of the kinds of information it allows researchers to register in their CV. This includes published journal and conference papers, advisements (both finished and ongoing), thesis committees, research projects, honors and distinctions, community service, affiliations, among others. The curriculum of every Brazilian researcher (Lattes CV) is stored in this platform and is publicly available through an HTML web page. Additionally, the Lattes Platform exports CV in the XML format, which can then be used as input for various automatic analyses.

The importance of Lattes CV has grown significantly. Today, it is the largest source of information regarding Brazilian researchers, students, managers, and other stakeholders of the national system of science, technology, and innovation community. Table 1 illustrates this fact[2] by showing the total amounts of Lattes CV aggregated by the type.

The big numbers on Table 1 raise a critical question: is this data consistent? It is well known that the ambiguity in the context of bibliographic citations is a problem of universal amplitude, which affects the quality of services and the content of digital libraries and similar systems [8]. Therefore, by analogy, it is possible to assume that these referential inconsistencies may have similar characteristics and effects in the Lattes Platform.

* Correspondence: vanessa@ic.uff.br
[2]Institute of Computing, Fluminense Federal University (UFF), Niterói, Brazil
Full list of author information is available at the end of the article

Table 1 Totals of Lattes curricula

Type	# of Lattes Curricula
Researcher with Ph.D. degree	171,869
Researcher with Master degree	308,196
Student	1,119,820
Other	1,164,656
Total	2,764,541

Since each researcher has her/his own Lattes CV, a given paper with more than one author must be present in the Lattes CV of each of its coauthors. Thus, there is a possibility that citations of journal and conference papers contain unconformities when compared to the curricula of their coauthors. The effect of these unconformities is especially challenging since CNPq and other funding agencies use this data to judge who gets research grants and who does not. As an example, assume that John inserts in his CV three papers that were supposedly coauthored by Mary and Lucas. When we go to Mary's and Luca's CV, however, we find only one of the papers. In this case, there is a referential inconsistency with two possible explanations: Mary's and Lucas's CV may be outdated, or there is an error in John's CV (may it be intentional or not). Thereby, if a funding agency uses this data to judge a grant proposal, the decision may be biased.

In this paper, we investigate this issue. Given a dataset *C* of curricula, our goal is to answer two research questions. (Q1) Is it possible to find inconsistencies in *C* based on the comparison of coauthor's list of publication? (Q2) Assuming the answer to Q1 is positive, how to determine the level of inconsistency in *C*? To answer question Q1, we propose and apply similarity-based heuristics to find inconsistencies in a large dataset of Lattes CV from researchers and professors of a Brazilian university. Then, to answer question Q2, we draw a map of referential inconsistencies identified in coauthoring of journal and conference papers on this dataset. Our main contribution resides in the heuristics, together with the inconsistency map. The heuristics are generic enough to be applied to other CV datasets, and thus are not specific for the Lattes Platform.

The remaining of this paper is organized as follows: We start by introducing background and discussing related work. The Research design and methodology section presents our matching strategy. In the Results and discussion section, we use the proposed heuristic algorithms over a real dataset composed of 2147 Lattes CV from professors of a Brazilian university to answer our two research questions. Finally, in Conclusions and Future Work section we conclude and suggest some future work.

Background and related work

There are two main problems involved in the process of identifying inconsistencies in datasets of electronic CV. Assuming we are analyzing a given CV *c*, for each paper *p* listed in *c*, we need to find the coauthors of *c* and retrieve their CV from the CV dataset. The second problem resides in, given two CV of coauthors, finding *p* in each of them. The next step is then to determine if there are inconsistencies. In the literature, these two problems have been intensively investigated.

The first one (matching author names) is directly related to the problem of disambiguation. There are two features of citation records that make this problem hard: homonyms (when there are multiple authors with the same name) and synonyms (when there are different (citation) names for the same author).

DBLP[3] uses a similarity function and a graph of coauthors to perform author disambiguation [20]. DBLP's disambiguation algorithm first builds a graph of coauthorship, where nodes represent author names and edges represent coauthorship. Then, nodes that are at a distance two in this graph are compared using the Levenshtein similarity distance. Very similar nodes are then checked more carefully since they are candidates for being synonyms [20]. For detecting homonyms, it uses a simple heuristic. When a new person comes into the digital library, and her/his name is already indexed by DBLP, then the list of their coauthors is checked. The person is then said to be the same as the one that shares some of the coauthors. If no common coauthor is found, then this person is treated as a new entry in DBLP (a new person) [19]. Other digital libraries such as MEDLINE[4] [32] and PubMed[5] [21] follow a similar path.

Several authors have worked on this problem as well [8]. They use a mix of techniques. While some use similarity functions [2, 7, 12, 18, 21, 27, 30], others use learning techniques [1, 14, 16, 28, 32, 35], heuristics [17, 19, 20, 24], classifiers [9, 10, 34] and clustering methods [11, 31].

To solve the second problem (matching publications), authors use disambiguation and deduplication methods. Deduplication methods use the semantics of metadata to provide good-quality results [3]. For scientific papers in digital libraries, the most widely used are those representing the authors and the title of the digital object. Borges et al. [3] present a comparative analysis of some classification algorithms (Naïve Bayes, Ripper and C4,5) and discuss an approach that combines similarity functions and classification algorithms to identify duplications in bibliographic metadata records.

Sarawagi and Bhamidipaty [26] argue that the main challenge of the deduplication process is to find a function able to distinguish when two records refer to the same entity in spite of possible errors and inconsistencies in the data. They address this issue by proposing a deduplication system based on machine learning called ALIAS. The central idea is to simultaneously create several redundant similarity functions, and then to exploit the differences between them to discover new types of inconsistencies between duplicates in a given database. Carvalho et al. [5] combine evidences provided by digital libraries to infer a similarity function at the record level using genetic programming. This function should be able to tell whether two records are replicas or not. It derives from a combination of weighted similarity functions applied at the field level.

Borges et al. [4] present an unsupervised heuristic approach aimed at bibliographic metadata deduplication, which devotes special attention to fields that refer to the names of authors to correctly identify redundancies in these metadata records. The process begins with a mapping of the metadata fields represented in different patterns. The focus of this approach is to build similarity functions specially developed for the domain of digital libraries. Metadata records are compared using these selected functions according to the domain of each attribute.

Yang et al. [35] focus on two kinds of correlations between citations. The first, named *Topic Correlation*, assumes that every researcher works on few research topics, and each of her/his publications is related to those topics. Thus, it measures the similarity between topics of two citations using implemented metrics from similarity functions, such as the *Cosine Similarity Metric* and the *Modified Sigmoid Function*. The Cosine Similarity Metric is used to estimate the similarity between two vectors, where each one represents the title attribute of the article. The *Modified Sigmoid Function*, on the other hand, relies on the co-occurrence of characteristics in two sets of corresponding attributes. The second correlation, named *Web Correlation,* is based on the premise that citations of a given researcher are generally listed in his publications Web page, or on the publications Web page of his coauthors. So if two citations co-occur in a Web page, they are probably related to the same researcher. Therefore, the Web Correlation means the cooccurrence's frequency of two citations on Web pages.

Almost all of the strategies discussed here for matching papers combine the use of machine learning, heuristics and similarity functions to obtain the best results. Although these techniques are quite effective, the environment they were designed for is much more complex than ours. They usually work over citation records found on the Web. This involves different representation of the citations together with several different sources of information. The Lattes CV scenario, on the other hand, is much less complex in the sense that there is a single information source involved, and all objects (papers) have the same structure.

Additionally, none of these approaches calculate the inconsistency level of the dataset of publications. In fact, inconsistencies that can happen in digital libraries are mainly caused by errors in the disambiguation algorithms. This is because most digital libraries are curated: there is a curator that is responsible for accepting new data records. Usually, those records are provided by journal editors or conference chairs, as it is the case of DBLP. This directly implies an important property: there are no duplicates of publications. The main problem is, thus, author disambiguation, that is, when a new publication record arrives, the problem is how to link it to the correct set of authors. On a dataset of Lattes CV, on the other hand, each author is responsible for inserting her/his publications, and this opens a large margin for inconsistencies. This is why, in this paper, we tackle the open problem of measuring the inconsistencies of Lattes datasets.

Since our data records are homogeneously structured, we use heuristics together with a similarity function. There are several similarity algorithms in the literature [1–3, 14] that explore different features of the data. Some are text-based, such as the Longest Common Subsequence (LCS) [15, 22, 33], Edit Distance [23] and the Jaccard Similarity Coefficient [13]. Others require that complex objects be represented as vectors (with values for several attributes), and use the cosine function to measure the distance between two given vectors [35]. In our work, we used the LCS [6] algorithm to calculate the similarity. This is a classical algorithm, widely used in several areas of scientific knowledge. Given two input sequences *s1* and *s2*, its goal is to identify the largest common sequence of characters between *s1* and *s2*. This is not, however, a similarity score that reflects the similarity of *s1* and *s2*. Instead, it is the size (measured in characters) of the largest common subsequence of *s1* and *s2*. To calculate the similarity, we multiply the LCS of *s1* and *s2* by 2 and divide the result by the sum of the sizes of *s1* and *s2*.

$$similarity(s1,\ s2)\ =\ \frac{2 * lcs(s1,\ s2)}{length(s1) + length(s2)}$$

When *s1* and *s2* are equal, *lcs(s1, s2)* is the size of *s1* (which is equal to the size of *s2* in this case), and thus the similarity is 1. When *s1* and *s2* have nothing in common, then *lcs(s1, s2)* is zero, and consequently, the similarity is also zero. In the next section, we describe how we use LCS to measure the inconsistency of a CV dataset.

Research design and methodology

To find inconsistencies in a dataset C of Lattes CV, we need to analyze each CV $c \in D$. For each paper p listed in c, we need to find the coauthors of p, and retrieve their CV $\{c_1, \ldots, c_n\}$ from the CV dataset. Then, we need to analyze each pair of CVs $(c, c_1), \ldots, (c, c_n)$ and find p in each of these pairs. For matching authors and papers, we use heuristics, which are described in this section. We start by investigating the best attributes to use in the matching, and then we present our heuristics. Since our heuristic uses a similarity function, we also perform a sensitivity test to choose an adequate threshold for this function.

Finding attributes to match authors and papers

In this work, we focus on the segments that are more challenging regarding checking for consistency amongst coauthors: published conference and journal papers. Table 2 provides a list of attributes the Lattes Platform provides for describing authors, conference papers and journal papers. In the table, multivalued attributes are marked with (*), and attributes that uniquely identify an entity are marked in *italic*.

Given this data, our initial challenge was to find answers to the following questions: (a) which attributes should we use to compare two author names when finding the list of coauthors of a paper?; and (b) which attributes should we use to compare two papers taken from two coauthors CV? The biggest problem when answering these questions is that there may be inconsistencies in virtually all attributes that describe a paper, and in some that describe an author.

To have an overview of the situation, we analyzed a dataset C_1 consisting of 58 Lattes CV of professors from a Computer Science department of a Brazilian university, looking specifically at the journal papers they have published. In total, this dataset contains 679 journal papers. Note that there are redundancies in these papers, since one paper usually appears in all of its coauthors' CV, so 679 is the sum of all papers cited in the 58 Lattes CV of C_1. We then tried to manually match these papers whenever the coauthors' CV was in our dataset. Twenty-four (24) of the fifty-eight (58) Lattes CV we analyzed presented some referential inconsistency (41.38%), totaling 103 inconsistencies of type "Reference errors in coauthoring citation" and 46 inconsistencies of type "Reference errors in journal papers." "Reference errors in journal papers" occurs when the journal paper from a particular CV is not found in the CV of one or more of its coauthors, while "Reference errors in coauthoring citation" occurs when the name of a coauthor in a given journal paper of a given CV cannot be found or is different from the name for citations informed in that coauthor's CV.

Table 2 Attributes that describe authors, conference papers, and journal papers

Entity	Attributes
Author	author_name, *lattes_id*, citation_name(*),
Conference paper	title, conf_name, isbn, *doi*, volume, first page, last page, year, coauthors(*)
Journal paper	title, jounal_name, issn, *doi*, volume, first page, last page, year, coauthors(*)

Note that 306 of the 679 journal papers of this dataset had one or more coauthors within the 58 professors selected by this study. The 103 inconsistencies of type "Reference errors in coauthoring citation" correspond to 33.66% of these 306. This means that 33.66% of the papers were not listed in some of their coauthors CV. In the same way, 46 inconsistencies of type "Reference errors in journal papers" represent 15.3%. These percentages are considerably large.

Another problem is the high percentage of null values for some of these attributes. The histogram of Fig. 1 illustrates this issue. It shows the percentage of null values of attributes that describe journal and conference papers using a small dataset C_2 containing 107 Lattes CV. This dataset contains CVs of professors of a Computing Graduate Program of a Federal University in Brazil ($C_1 \subset C_2$), together with CVs of some of their most prolific coauthors from other departments and universities. This dataset contains 1976 journal papers and 7096 conference papers. These numbers correspond to the sum of all journal papers and conference papers, respectively, cited in the 107 Lattes CVs of C_2.

The DOI identifier, for instance, would be an excellent identifier to match papers in the CV of two coauthors, since it is unique. However, due to the significant amount of null values it presents (it is null in 56.78% of the journal papers and 91.98% of the conference papers), it is not a good choice to be used to match papers. The same happens with the ISBN of conference papers (78.59% of null values). The publication year, paper title, and journal/conference name, on the other hand, are good attributes, since they are always present in the dataset. In fact, these are mandatory attributes for both conference and journal papers in the Lattes Platform. This does not mean, however, that they are exactly the same in all coauthors' curricula, and this needs to be taken into account when matching two papers.

Based on this preliminary analysis, we have decided to check attribute matches in the following order when matching conference and journal papers: title, volume, year, initial page number, final page number, DOI, ISSN (for journal papers), ISBN (for conference papers), journal name (for journal papers), conference name (for conference papers), and the coauthor

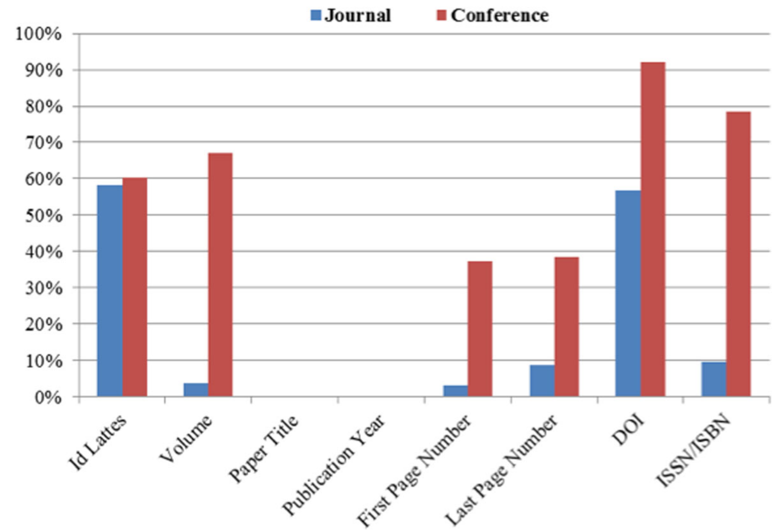

Fig. 1 Histogram of null values in journal and conference papers

names. As for matching authors, we use Lattes ID, full name, and citation name, in this order.

As a result of this analysis, the following attributes were identified as possible sources of inconsistencies: paper title, volume, year, first page number, last page number, DOI, ISSN (for journal papers), ISBN (for conference papers), journal name (for journal papers), conference name (for conference papers), and the order of authorship.

As mentioned in the Background and related work section, we use similarity functions to compare authors and papers. The thresholds we use to define when two authors or papers should be treated as a match are named *tCoauthor*, *tJournalTitle*, and *tConferenteTitle*. When two coauthor names *a1* and *a2* are compared, we consider they correspond to the same coauthor if *similarity(a1, a2) > = tCoauthor*. The same holds for paper titles (using tConferenteTitle or tJournalTitle, depending on the type of the publication). However, the similarity is only used when we do not find the corresponding entity by a direct (equality) comparison. We thus use a heuristic that we explain in the next section.

Heuristic matching algorithm

Assume a dataset *C* of Lattes CV where $C = \{c_1, c_2, \ldots, c_n\}$. Each c_i in *C* is composed of an author name (*author_name*), a Lattes ID (*lattes_id*), a set of citation names (*cn*) for that author (for instance, the citation names for John Mitchel Smith may be SMITH, J.; SMITH, J.M.), and a set of conference (*cp*) and journal papers (*jp*). Thus, c_i can be described as $[author\_name, lattes\_id, \{cn_1, \ldots, cn_o\}, \{cp_1, \ldots, cp_p\}, \{jp_1, \ldots, jp_q\}]$. Also, each conference paper cp_j is composed by a title, conference name, ISBN, DOI, volume, first page number, last page number, year, and a set of coauthors, thus $cp_j = [title, conf\_name, isbn, doi, volume, first\_page, last\_page, year, \{[ca_1, lattes\_id\_ca_1], \ldots, [ca_r, lattes\_id\_ca_r]\}]$. Note that, in this case, the coauthors' names may vary: they may be the complete name or a citation name. In some cases, the Lattes ID of a coauthor is provided associated with the coauthor's name. Journal papers are described by a similar structure: $jpk = [title, journal\_name, issn, doi, volume, first\_page, last\_page, year, \{[ca_1, lattes\_id\_ca_1] \ldots, [ca_r, lattes\_id\_ca_r]\}]$.

Our heuristic matching algorithm processes each *c* in *C* individually, as well as each *cp* and *jp* in *c*. For each conference and journal paper, the processing starts by trying to find the CV of the coauthors, as summarized in Algorithm 1. Notice that, in the heuristic, the input parameter *lattes_id* may be null, since this value may not be present in *cp* or *jp*.

Initially, we try to retrieve the coauthor by Lattes ID (lines 1–4 of Algorithm 1). Note that this is an exact match, so this query may return a single tuple (which is returned by the algorithm at line 4), or no tuple. If no tuple is returned found, we attempt to retrieve the coauthor by the full name (lines 6–7), which is returned in case it is found (line 9). If there is no success and if the coauthor's name contains a period and/or a semicolon (which characterizes a citation name) (line 10), the heuristic tries to find the coauthor's CV by using the citation name (lines 11–14). Since this query may retrieve several tuples, we use a similarity function to find the most similar coauthor in the database (lines 15–24).

Once we have found the coauthor's CV, we proceed to search for the publication (referred as the *target publication*) in it. The heuristic algorithm that performs this task for journal papers is summarized in Algorithm 2. The heuristic for conference papers is analogous.

Algorithm 1 - Heuristic for matching coauthors

Find-Coauthor(*ca, lattes_id_ca, tCoauthor*)

```
s <- SELECT author_name, lattes_id FROM cv
     WHERE lattes_id = lattes_id_ca
if s is not empty
    return s[0].lattes_id
else
   s <- SELECT author_name, lattes_id FROM cv
        WHERE author_name = ca
   if s is not empty
        return s[0].lattes_id
if s is empty and (contains(ca, “;”) or contains(ca, “.”))
   s <- SELECT author_name, lattes_id
        FROM citation_names cn, cv
        WHERE cv.id = cn.cv_id
          AND cn.citation_name = ca
if s is not empty
    maxsim <- 0
    maxt <- null
    for each t in s
        sim <- similarity(ca, t.author_name)
        if sim ≥ maxsim
            maxsim <- sim
            maxt <- t
    if maxsim ≥ tCoauthor
        return maxt.lattes_id
return null
```

In several cases, we cannot locate a given publication in the coauthor's CV due to differences in the title spelling (lines 1–3 of Algorithm 2). In this case, our heuristic attempts to retrieve it by using a set of attributes: the Id of the coauthor's CV, year of publication, volume, and number of first and last pages (lines 5–10). If the publication is not located, we use the most stable attributes, which are the id of the coauthor's CV and the publication year (lines 12–14). If we still could not find it, two last attempts are made before trying a similarity approach. Both of them try to locate the publication in the other segment (if it is a journal, it looks into conference papers, and vice-versa). The first attempt uses only the publication title (lines 16–17), and finally the id of the coauthor's CV and the publication year (lines 19–21). The queries we use in lines 12–14, 16–17, and 19–21 are not exact, and thus, in this case, we apply a similarity function for each publication retrieved by these queries (lines 22–31). The most similar title (when the similarity is above the *tJournalTitle* or *tConferenceTitle* thresholds, depending on the publication type) is assumed to be the correct one. Once we have the target publication and the corresponding publication in the coauthor's CV, we proceed to compare their attributes to identify inconsistencies.

Sensitivity test

Since our heuristics depend on three different thresholds (*tCoauthor*, *tJournalTitle*, and *tConferenteTitle*), we executed a sensitivity test to define the best thresholds. Our goal was to obtain thresholds that would minimize the occurrence of false positives and false negatives.

Han, Kamber, and Pei [13] advocate the use of precision and recall for an effective search for optimal similarity thresholds. The combination of the highest precision with the highest recall provides the highest effectiveness. This combination is reflected by the harmonic mean and is known by F-Measure. Therefore, the most effective similarity thresholds would be those with the highest F-Measure.

The precision and recall measures require a well-defined comparison set so that we know the correct equivalences. In other words, we need to know exactly which paper *p* in a given CV corresponds to which paper *p'* on a coauthors CV. Thus, we built an oracle using a subset C_3 of our dataset C_2. This dataset C_3 contained

Algorithm 2 - Heuristic for matching journal papers

Find-Journal-Publication(*jp*, *tJournalTitle*)

```
s <- SELECT id FROM journalPaper WHERE title = jp.title
if s is not empty
    return s[0].id
else
    s <- SELECT j.id FROM journalPaper j, cv
         WHERE j.coauthor_cv = cv.id
           AND j.year = jp.year
           AND j.volume = jp.volume
           AND j.first_page = jp.first_page
           AND j.last_page = jp.last_page
if s is empty
    s <- SELECT id FROM journalPaper j, cv
         WHERE j.coauthor_cv = cv.id
          AND j.year = jp.year
if s is empty
    s <- SELECT id FROM conferencePaper j, cv
         WHERE title = jp.title
if s is empty
    s <- SELECT id FROM conferencePaper j, cv
         WHERE j.coauthor_cv = cv.id
          AND j.year = jp.year
if s is not empty
    maxsim <- 0
    maxt <- null
    for each t in s
        sim <- similarity(jp.title, t.title)
        if sim ≥ maxsim
            maxsim <- sim
            maxt <- t
    if maxsim ≥ tJournalTitle
        return maxt.id
return null
```

43 CV, totaling 1026 journal papers and 3053 conference papers, which correspond to the sum of all journal and conference papers cited in those 43 CV. To build this oracle, our heuristic algorithm was executed using very low values for the thresholds, and then all the matches were manually classified as positives or negatives, depending on the fact that the matching was correct or incorrect. In the end, our oracle was composed of two tables: one for true positives and another for true negatives.

We then generated the sensitivity matrix by running an experiment where the oracle was accessed to confirm the matches found by our heuristic algorithm. This test considered the interval [0.40 to 0.90] for each similarity threshold and a variance of 0.05 for each execution cycle.

Table 3 Sensitivity Matrix for Journal Papers

tJournalTitle	tCoauthor	True positives	False positives	True negatives	False negatives	Precision	Recall	F1-measure
0.55	0.40	349	6	1889	27	0.983099	0.928191	0.954856
0.55	0.45	349	6	1889	27	0.983099	0.928191	0.954856
0.55	*0.50*	*349*	*6*	*1889*	*27*	*0.983099*	*0.928191*	*0.954856*
0.55	0.55	348	6	1890	27	0.983051	0.928000	0.954733
0.50	0.40	349	7	1888	27	0.980337	0.928191	0.953552

Table 4 Sensitivity matrix for conference papers

tConferenceTitle	tCoauthor	True positives	False positives	True negatives	False negatives	Precision	Recall	F1-measure
0.65	*0.55*	*1192*	*96*	*6784*	*131*	*0.925466*	*0.900983*	*0.913060*
0.55	0.55	1205	119	6764	115	0.910121	0.912879	0.911498
0.60	0.55	1194	108	6773	128	0.917051	0.903177	0.910061
0.65	0.60	1181	96	6784	142	0.924824	0.892668	0.908462
0.50	0.50	1219	148	6738	98	0.891734	0.925588	0.908346

Tables 3 and 4 show part of the sensitivity matrix for journal and conference papers, respectively. According to our results, in the context of journal papers, the values that maximize F-Measure are 0.55 for *tJournalTitle* and 0.5 for *tCoauthor*. For conference papers, the best values are 0.65 for *tConferenceTitle* and 0.55 for *tCoauthor*. These are the values we used to build our inconsistency map.

Results and discussion

With the best values of the thresholds of similarity in hand, we proceeded to analyze a larger dataset C_4. This dataset C_4 is composed of 2147 Lattes CV from professors of a Brazilian university. This dataset contains 242,333 coauthors, 32,697 journal papers, and 62,144 conference papers, which correspond to the sum of all journal and conference papers cited in those 43 CV. Our main goal with this evaluation is to measure the quality of this dataset, trying to answer the questions raised in the introduction of this paper: (Q1) is it possible to find inconsistencies in C_4 based on the comparison of coauthor's lists of publication? (Q2) assuming the answer to Q1 is positive, how to determine the level of inconsistency in C_4?

The totals in Table 5 indicate how the coauthors were matched in the dataset. Note the low percentage of coauthors matched by name. This fact can be justified for two reasons. The first one is the fact that a significant part of coauthors belong to other institutions, and thus our dataset does not contain their Lattes CVs. The second probable reason is the low quality of the data.

Despite the low percentage of coauthors matched by citation name, this kind of matching can be considered relatively effective for conference papers. Note that this indicates a semantic distortion in the data because the citation name is informed in the publication instead of the complete name of coauthors.

A global perception of the results indicates that we could successfully match 18.51% of the coauthors. This is consistent with the interpretation that a significant portion of coauthors belongs to other institutions, and thus their CVs are not in our database. This is reinforced by the low percentage of coauthors retrieved by Lattes identifier. It is important to point out that we proceed with the comparison of publications only when the coauthor's CV is located.

Table 6 indicates how the 32,697 journal papers and 62,144 conference papers in C_4 were matched. It is possible to observe that matching by title was more effective for journal papers than for conference papers. Due to this fact, we can conclude that titles of journal papers have more quality than titles of conference papers in our dataset. Curiously, Brazilian funding agencies have an evaluation procedure that takes only journal papers into account for several research areas. This may explain the better quality of this data. Another interesting point regarding matching of publications resides in the fact that a larger amount of matches of journal papers were made by title instead of by similarity, which corroborates with our hypothesis of better data quality in this segment. We must emphasize, however, the low percentage of matches (26.14%), which is justified by the low frequency of publications among coauthors within the same institution.

Map of inconsistencies

With the matching in hand, it was possible to produce a map of inconsistencies. Table 7 shows the results. Note that we used the same dataset C_4 of 2147 CV. Note also that four of these CV do not contain conference papers

Table 5 Matches of coauthors

Locating method	Analyzed segment		
	Journal papers	Conference papers	Total
Coauthors	87,028	155,305	242,333
Matched by Lattes identifier	11,632 (13.37%)	16,842 (10.84%)	28,474 (11.75%)
Matched by Name	2649 (3.04%)	8495 (5.47%)	11,144 (4.60%)
Matched by citation name	1658 (1.91%)	3582 (2.32%)	5240 (0.22%)
Total	15,939 (18.31%)	28,919 (18.62%)	44,858 (18.51%)

Table 6 Matches of publications

Locating method	Analyzed segment		
	Journal papers	Conference papers	Total
Publications	32,697	62,144	94,841
Matched by title	7357 (22.50%)	8644 (13.91%)	16,001 (16.87%)
Matched by similarity	5894 (18.03%)	2893 (4.66%)	8787 (9.26%)
Total	13,251 (40.53%)	11,537 (18.56%)	24,788 (26.14%)

(thus, 2143 CV were analyzed for inconsistencies in conference papers). To calculate the numbers in Table 7, we take, for a given publication *p* in a given CV *c*, all the coauthors of *p*. Assuming that *p* has *n* coauthors, but only *m* ($n \geq m$) were found in our dataset, we try to find *p* in the CV of these *m* coauthors. Then, we count the inconsistencies comparing each one of the m CV with c, adding one to each inconsistency category as needed. Thus, a single publication can contribute to more than one type of inconsistency, and even to the same type of inconsistency more than once (in the case the same inconsistency happens in the CV of two or more coauthors). According to this table, the most frequent inconsistencies were "Different Initial/Final Pages" and "Publications not Found in Coauthors" in journal papers and conference papers, respectively.

The total percentage of inconsistencies remains close in both segments. However, the journal paper segment shows a lower percentage of inconsistencies (17.70%). As mentioned previously, for some research areas, Brazilian funding agencies have an evaluation procedure that takes only journal papers into account. This may explain the better quality of the data in this segment.

The percentage of inconsistencies of type "Publication found in another segment" can be considered relevant, despite its low value (0.53%). Note that this fact highlights a divergence in the classification of some publications and can distort the generation of productivity indicators in a given segment.

The total percentage of publications with inconsistencies (18.98%) can be considered high for a dataset consisting of professors of a Federal University. We note that these CV are used by CNPq and other funding agencies for distributing research grants.

Conclusions and future work

This paper detailed an approach to find inconsistencies in a set of electronic CV using a similarity-based approach. Using this approach, we were able to answer two research questions as follows:

Q1: Is it possible to find inconsistencies in a dataset *C* of Lattes CV based on the comparison of coauthor's list of publication?

Answer: Yes, it is. Using our heuristics, we were able to find several types of inconsistencies in a large dataset of Lattes CV, which include publication not found in coauthors CV, publication found in another segment (for example, a journal paper found as conference paper in the coauthors dataset), journal/conference paper not found in a coauthor CV, different order of

Table 7 Total of inconsistencies in the institution

Inconsistency	Analyzed segment		
	Journal papers	Conference papers	Total
CV	2147	2143	2147
Coauthors	87,028	155,305	242,333
Publication not found in coauthors	2193 (2.52%)	10,847 (6.98%)	13,040 (5.38%)
Publication found in another segment	495 (0.57%)	781 (0.50%)	1276 (0.53%)
Journal/conference title not found	980 (1.13%)	5127 (3.30%)	6.107 (2.52%)
Different order of coauthorship	2556 (2.94%)	4840 (3.12%)	7396 (3.05%)
Different year of publication	318 (0.37%)	679 (0.44%)	997 (0.41%)
Different volume number	2289 (2.63%)	4788 (3.08%)	7077 (2.92%)
Different initial/final pages	4623 (5.31%)	3237 (2.08%)	7860 (3.24%)
Different DOI	107 (0.12%)	5 (0.00%)	112 (0.05%)
Different ISSN/ISBN	1840 (2.11%)	288 (0.19%)	2128 (0.88%)
Total	15,401 (17.70%)	30,592 (19.70%)	45,993 (18.98%)

coauthorship in a given publication, different year of publication, different volume number, different initial/final pages, different DOI, and different ISSN/ISBN.

Q2: Given a dataset *C* of Lattes CV, how to determine its level of inconsistency?

Answer: To answer this question, we proposed the creation of a map of inconsistencies and demonstrated it using a dataset of 2147 Lattes CV. The obtained results indicate the existence of 18.98% of referential inconsistencies. We believe this is a significant amount in a dataset that is supposed to be correct and trustable.

The main goal of this study is to attract the attention of the research community to the importance of maintaining the reliability of curricula data. This data is very useful not only for evaluation of bibliographic production as well as for the generation of bibliometric indicators.

Our approach is generic in the sense that it can be applied to any electronic CV dataset. In fact, any dataset where the input is made by the researcher instead of a controlling organization may suffer from inconsistencies. In such cases, the heuristics can be adapted to use the attributes provided by the dataset. Lattes ID, for example, would not exist, but there may be an attribute that plays the same role. If not, it would be a matter of simply removing this step from the heuristics and using the remaining ones. On the other hand, we believe that datasets like DBLP and similar systems do not suffer from this kind of inconsistency since a paper is inputted a single time in the dataset and linked to each researcher profile. There is no way for the researcher herself/himself to modify, include, or exclude data from the DBLP dataset.

As future work, we plan to improve our integrity check by using weights to indicate the frequency in which an author publishes with a particular coauthor. That is, the inconsistencies checked at very frequent coauthors would receive a higher weight than those associated with less frequent coauthors. We also plan to classify the inconsistencies according to a severity criterion.

Another interesting future direction would be to use metadata provided by curated sources like DBLP, ACM Digital Library, IEEE Explore, among others, to check for inconsistencies. This way, instead of finding the coauthor Lattes CV, we would be able to directly compare the publication titles. In other words, for every publication *p* in the CV of an author *a*, we would try to find *p* in a curated digital library. Then, the inconsistency checking would be made using the attributes of *p* in *a*'s CV and in the digital library. We believe this would increase the precision of our results since this would not depend on the coauthors having a Lattes CV. However, this would require a previous step of mapping attributes of the digital library. This could be done manually, since the number of attributes is small, or using any existing schema matching approach [25, 29]. After the mapping is done once, it can be reused to compare several publications.

Finally, we could invest in a monitoring service aiming at reducing inconsistencies. The service would continuously watch a given Lattes CV. It would be the author's choice to have her/his Lattes monitored by providing the Lattes URL and an email address. Then, every time the service detects the inclusion of a new publication in a monitored CV, it would apply the heuristics to check for inconsistency. In case it finds some, it would warn the author by sending her/him an email with the details. We believe this would contribute to reducing inconsistencies since the author would be immediately notified and be able to make the corrections. For the cases where the publication is not found in the coauthor's CV, the service could send the coauthors an email, alerting about a new publication, and asking them to include it in their CV. This would work for other authors that use the watching service, or the original author could provide the email addresses of her/his most frequent coauthors. This service could also work using metadata provided by curated sources, as previously explained.

Endnotes

[1] http://lattes.cnpq.br/

[2] Source: http://estatico.cnpq.br/painelLattes/mapa/

[3] http://dblp.uni-trier.de

[4] https://www.nlm.nih.gov/pubs/factsheets/medline.html

[5] https://www.ncbi.nlm.nih.gov/pubmed/

Acknowledgements
This work was partially funded by CNPq and FAPERJ.

Authors' contributions
IR and VB participated in the conception of the approach and on the related work analysis. IR implemented the approach and run the experiments. VB helped on analyzing the experimental results and writing and polishing the paper. This work was conducted while IR was at UFF. All authors reviewed the paper and agree with this submitted version.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] NCE, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil.
[2] Institute of Computing, Fluminense Federal University (UFF), Niterói, Brazil.

Received: 23 September 2015 Accepted: 9 February 2017
Published online: 23 February 2017

References

1. Aron Culotta, Pallika Kanani, Robert Hall, Michael Wick and Andrew McCallum (2007) Author disambiguation using error-driven machine learning with a ranking loss function. International Workshop on Information Integration on the Web (IIWeb), Vancouver
2. Bhattacharya, I. and Getoor, L (2007) Collective entity resolution in relational data. ACM Trans Knowledge Discovery Data. 1(1):1–36
3. Borges EN, Becker K, Heuser CA, Galante R (2011) A classification-based approach for bibliographic metadata deduplication. WWW/Internet International Conference, Porto, pp 221–228
4. Borges EN, Carvalho MG, Galante R, Gonçalves MA, Laender AHF (2011) An unsupervised heuristic-based approach for bibliographic metadata deduplication. Inf Process Manage 47:706–718
5. Carvalho MG, Gonçalves MA, Laender AHF, Silva AS (2006) Learning to deduplicate. ACM/IEEE-CS Joint Conference on Digital libraries, Chapel Hill, pp 11–15
6. Cormen, TH, Leiserson, CE, Rivest, RL and Stein, C (2009) Introduction to algorithms. The MIT Press, Cambridge
7. Cota RG, Ferreira AA, Nascimento C, Gonçalves MA, Laender AHF (2010) An unsupervised heuristic-based hierarchical method for name disambiguation in bibliographic citations. J Am Soc Inf Sci Technol 9:1853–1870
8. Ferreira AA, Gonçalves MA, Laender AHF (2012) A brief survey of automatic method for author name disambiguation. SIGMOD Record 41:15–26
9. Ferreira AA, Veloso A, Gonçalves MA, Laender AHF (2010) Effective self-training author name disambiguation in scholarly digital libraries. Annual Joint Conference on Digital Libraries (JCDL), New York, pp 39–48
10. Han H, Giles L, Zha H, Li C, Tsioutsiouliklis K (2004) Two supervised learning approaches for name disambiguation in author citations. ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL), New York, pp 296–305
11. Han H, Xu W, Zha H, Giles CL (2005) A hierarchical naive Bayes mixture model for name disambiguation in author citations. ACM Symposium on Applied Computing (SAC), New York, pp 1065–1069
12. Han H, Zha H, Giles CL (2005) Name disambiguation in author citations using a K-way spectral clustering method. ACM/IEEE-CS Joint Conference on Digital Libraries, New York, pp 334–343
13. Han J, Kamber M, Pei, J (2012) Data mining concepts and techniques. Morgan Kaufmann, Burlington
14. Huang J, Ertekin S, Giles CL (2006) Efficient name disambiguation for large-scale databases. In: Fürnkranz J, Scheffer T, Spiliopoulou M (eds) Knowledge Discovery in Databases: PKDD 2006. Springer, Berlin Heidelberg, pp 536–544
15. Hunt JW, Szymanski TG (1977) A fast algorithm for computing longest common subsequences. Commun ACM 20(5):350–353
16. Kanani P, McCallum A, Pal C (2007) Improving author coreference by resource-bounded information gathering from the web. International Joint Conference on Artifical Intelligence, San Francisco, pp 429–434
17. Kang I-S, Na S-H, Lee S, Jung H, Kim P, Sung W-K, Lee J-H (2009) On co-authorship for author disambiguation. Inf Process Manage 45(1):84–97
18. Lee D, On B-W, Kang J, Park S (2005) Effective and scalable solutions for mixed and split citation problems in digital libraries. International Workshop on Information Quality in Information Systems (IQIS), New York, pp 69–76
19. Ley M (2009) DBLP: some lessons learned. Proc VLDB Endow 2(2):1493–1500
20. Ley M, Reuther P (2006) Maintaining an online bibliographical database: the problem of data quality. Journées Extraction et Gestion des Connaissances (EGC), Lille, pp 5–10
21. Liu W, Islamaj Doğan R, Kim S, Comeau DC, Kim W, Yeganova L, Lu Z, Wilbur WJ (2014) Author name disambiguation for PubMed. J Assoc Inf Sci Technol 65(4):765–781
22. Masek WJ, Paterson MS (1980) A faster algorithm computing string edit distances. J Comput Syst Sci 20(1):18–31
23. Navarro G (2001) A guided tour to approximate string matching. ACM Comput Surveys 33(1):31–88
24. Pereira DA, Ribeiro-Neto B, Ziviani N, Laender AHF, Goncalves MA, Ferreira AA (2009) Using web information for author name disambiguation. ACM/IEEE-CS Joint Conference on Digital libraries, Austin, pp 49–58
25. Rahm E, Bernstein PA (2001) A survey of approaches to automatic schema matching. VLDB J 10(4):334–350
26. Sarawagi S, Bhamidipaty A (2002) Interactive deduplication using active learning. ACM SIGKDD International Conference on Knowledge discovery and data mining, Edmonton, pp 269–278
27. Shin D, Kim T, Choi J, Kim J (2014) Author name disambiguation using a graph model with node splitting and merging based on bibliographic information. Scientometrics 100(1):15–50
28. Shu L, Long B, Meng W (2009) A latent topic model for complete entity resolution. IEEE International Conference on Data Engineering (ICDE), Washington, pp 880–891
29. Shvaiko, P and Euzenat, J (2005) A survey of schema-based matching approaches. Journal on Data Semantics IV. S. Spaccapietra, ed. Springer Berlin Heidelberg. 146–171.
30. Song Y, Huang J, Councill IG, Li J, Giles CL (2007) Efficient topic-based unsupervised name disambiguation. ACM/IEEE-CS Joint Conference on Digital Libraries, New York, pp 342–351
31. Tang J, Fong ACM, Wang B, Zhang J (2012) A unified probabilistic framework for name disambiguation in digital library. IEEE Trans Knowledge Data Eng 24(6):975–987
32. Torvik, VI and Smalheiser, NR (2009) Author name disambiguation in MEDLINE. ACM Trans Knowl Discov Data. 3(3)11:1–11:29.
33. Ullman JD, Aho AV, Hirschberg DS (1976) Bounds on the complexity of the longest common subsequence problem. J ACM 23(1):1–12
34. Veloso A, Ferreira AA, Gonçalves MA, Laender AHF, Meira W Jr (2012) Cost-effective on-demand associative author name disambiguation. Inf Process Manage 48(4):680–697
35. Yang K-H, Peng H-T, Jiang J-Y, Lee H-M, Ho J-M (2008) Author name disambiguation for citations using topic and web correlation. In: Christensen-Dalsgaard B, Castelli D, Ammitzbøll Jurik B, Lippincott J (eds) Research and Advanced Technology for Digital Libraries. Springer Berlin, Heidelberg, pp 185–196