Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 901–910
Marseille, 11–16 May 2020

Deduplication of Scholarly Documents using Locality Sensitive Hashing and Word Embeddings

Bikash Gyawali, Lucas Anastasiou, Petr Knoth
Knowledge Media Institute
The Open University
Milton Keynes, United Kingdom
{bikash.gyawali, lucas.anastasiou, petr.knoth}@open.ac.uk

Abstract

Deduplication is the task of identifying near and exact duplicate data items in a collection. In this paper, we present a novel method for deduplication of scholarly documents. We develop a hybrid model which uses structural similarity (locality sensitive hashing) and meaning representation (word embeddings) of document texts to determine (near) duplicates. Our collection constitutes a subset of multidisciplinary scholarly documents aggregated from research repositories. We identify several issues causing data inaccuracies in such collections and motivate the need for deduplication. In lack of existing dataset suitable for study of deduplication of scholarly documents, we create a ground truth dataset of $100K$ scholarly documents and conduct a series of experiments to empirically establish optimal values for the parameters of our deduplication method. Experimental evaluation shows that our method achieves a macro F1-score of 0.90. We productionise our method as a publicly accessible web API service serving deduplication of scholarly documents in real time.

Keywords: Deduplication, Scholarly Documents, Locality Sensitive Hashing, Word Embeddings, Digital Repositories

1. Introduction

Publishing research findings as scholarly documents (publications) has always been the mainstream model for disseminating scientific research. To this end, authors publish their research outputs as scholarly documents and deposit them to one or more publishing platforms, *repositories*, of their choice. Such choices include institutional repositories, personal web pages, preprint services, academic publishers' platform and so on. Authors choosing to submit their article to multiple repositories have different motivations to do so, such as:

- Different versions of author's manuscript become suitable for submission to different repositories. Examples include, preprint repositories for submitting manuscripts that are yet to be peer reviewed, authors' personal web pages for hosting open access versions of their publications, etc.
- Institutional as well as national policies mandate authors submit their research outputs to their own institution's repository.
- For publications with multiple authorship, each author may decide to submit it to one or more repositories of their own choice/institution.
- By submitting the same research work to multiple different repositories, a wider audience can be reached (for example, documents from open access repositories are available to everyone, while the publisher might put the document behind a paywall); research can also be disseminated sooner thereby increasing accelerating the scholarly communication process.

While authors may submit **exact duplicate** copies of their research output to multiple repositories, they might also introduce slight variations (**near duplicates**) while submitting to different repositories (Klein et al., 2016). Throughout the remainder of this text, we will use the generic term **duplicates** to refer to both exact and near duplicates. For example, authors might submit revised versions (preprint, author's copy, camera-ready) of the same research article to different repositories over time and different formats of document submissions (e.g. pdf, LaTeX) are also prevalent. Repositories usually rely on authors manually entering metadata information for their articles during submission. This gives way to introducing errors, omissions and typos in document metadata; creating documents with corrupt or missing metadata across multiple repositories.

Such duplicates are of major concern to applications which target processing of scholarly documents aggregated from multiple repositories. Table 1 shows some example duplicates that can arise when aggregating documents from multiple repositories. Example A represents documents that are exact duplicates of each other but are, nevertheless, present in multiple repositories. Example B represents duplicates resulting from document formatting error while examples C and D represent duplicates arising from revisions, paraphrasing or updates to documents while they are submitted to different repositories. For text/data mining applications, these are redundant and/or inconsistent data which can skew data distribution and lead to an imbalanced dataset (Kołcz et al., 2003).

The task of identifying duplicates in a data collection is known as deduplication. In this paper, we present a novel deduplication method for scholarly documents. Many existing work on deduplication (e.g. (Chaudhuri et al., 2003), (Jiang et al., 2014)) describe methods for detecting **duplicate entities** (person, organization etc.) organised into databases/graphs and rely on direct matching of one or more attribute-value pairs (metadata) making up such data items. Other work (e.g. (Forman et al., 2005), (Bogdanova et al., 2015)) have discussed content based approaches for identifying **duplicate documents** and use similarity of tex-

Example	Source Repository	Document Content	Why duplicates?
A	Springer - Publisher Connector	Title = Profiling sugar metabolism during fruit ...	Exact same titles but documents aggregated from different repositories.
	ProdInra	Title = Profiling sugar metabolism during fruit ...	
B	Elsevier - Publisher Connector	Abstract = AbstractThe formation of smart, Metal Matrix Composite (MMC) structures through the use of solid-state ...	The abstracts are the same except for error introduced during document submission into different repositories.
	Loughborough University Institutional Repository	Abstract = This is an open access article under the CC BY license(http://\\ud\ncreativecommons.org/licenses/by/4.0/). The formation of smart, Metal Matrix Composite (MMC) structures through the use of solid-state...	
C	Swinburne Research Bank	Abstract = We present an analysis of ... 20-ms pulsars ...	Slight variation in text (20-ms vs 20 millisecond) on document versions on two different repositories.
	arXiv.org e-Print Archive	Abstract = We present an analysis of ... 20 millisecond pulsars ...	
D	Archivio della ricerca - Università degli studi di Napoli Federico II	Title = Simulation of Gaussian Processes and First Passage Time Densities Evaluation Abstract= Motivated by a typical and .... first passage time probability densities.	Possibly different paraphrasing of the title for the exactly same abstract; the duplicates can only be identified when comparing "Abstract" rather than "Title".
	Archivio della ricerca - Università degli studi di Napoli Federico II	Title = Vectorized simulations of normal processes for first-crossing-time problems Abstract = Motivated by a typical and ... first passage time probability densities.	

Table 1: Examples of duplicates in documents aggregated from multiple repositories

tual content of documents for the task. In particular, using document hash values for deduplication has been shown to be effective for deduplication of documents in specific collections (e.g. web corpus (Manku et al., 2007), clinical notes (Shenoy et al., 2017)) but similar study for deduplication of scholarly documents has not been reported so far. It is important that this study be carried out because i) scholarly collections have a number of issues related to data inaccuracies (see Section 2.) and therefore matching of attribute values cannot be reliably used ii) for scholarly documents, the only available content may often be short abstract text only (due to copyright issues) and iii) scholarly text is often technical in nature and has complex linguistic structure compared to general purpose text on the web. In this paper, we address this research gap and propose a hybrid method which takes into account different models of document content similarity for determining duplicates of scholarly documents. Namely, we build on top of matching structural similarity (using locality sensitive hash values) and meaning representations (using word embeddings) of documents' content for identifying duplicates.

We first construct a ground truth dataset labeling duplicates/non-duplicates in a collection of 100K scholarly documents aggregated from multiple different repositories and across scholarly disciplines. Next, we define separate deduplication methods based on different document similarity measures (locality sensitive hashing vs. word embeddings) and analyse their performance. Finally, we build a hybrid method which builds upon individual methods and empirically establish the best values for its parameters by conducting a series of experiments. We show that this method performs competitively towards correctly identifying duplicates (and non-duplicates) – a macro F1 score of 0.90 and an accuracy of 90.30% is obtained.

We expose our deduplication system as a web API implemented over a much larger collection (over 130 million scientific documents) of research outputs aggregated from multiple repositories world-wide. By enabling open access to this collection and exposing the deduplication API, we create a man/machine interface to the deduplication service which identifies duplicate documents that exist across repositories for a given scientific document at hand.
Our novel/main contributions are:

- We propose and evaluate different content based deduplication methods (locality sensitive hashing vs. word embeddings) and study their effectiveness in the context of deduplication of scholarly documents.
- We design a new hybrid method for deduplication which builds upon the strength of individual methods and improves the performance of scholarly documents' deduplication.
- We construct a ground truth dataset of scholarly documents for deduplication purposes and make it publicly available. There are no existing datasets of this nature which we are aware of.
- To our knowledge, we produce the first open API for finding duplicates of scientific documents in real time.

2. Problem Statement

In the context of scholarly documents, a reasonable approach to deduplication would seem to be matching document identifiers, especially the DOI, or other metadata information, such as the title or author names, associated with such documents. Repositories usually expose such information and make them available in a structured format e.g. xml suitable for automatic processing. However, deduplication approaches based on direct matching of such attributes would be far from ideal because they can't be reliably used for deduplication of scholarly documents across repositories. The Digital Object Identifier (DOI) is a common scheme used by publishers to give documents a unique identity. However, many documents with unassigned DOI exist such as those in preprint repositories. Likewise, repositories can expose erroneous DOIs to the documents they contain, for example, by using the generic DOI of a journal to all the articles within the journal.

In a collection of scholarly documents we considered (Section 4.1.), more than 82% of documents did not have a DOI and we observed that the most frequent DOIs in the collection were generic DOIs (e.g.: 10.4028/www.scientific.net, 10.1093/mnras). We identified the most frequent $1,000$ DOIs in our source collection with their frequencies of occurrence – ranging from 65 to $45,184$. Also, it is not clear if near duplicates will have the same DOI at all, especially when they are submitted across different repositories.

Similar problems appear with other metadata information such as document titles. For open access articles, OAI identifier is used as unique identifier of documents but it doesn't allow for detection of duplicates. We analysed the most frequent $8,500$ document titles in our source collection and observed that many had incorrectly assigned titles and with multiple occurrences (ranging from 96 to $549,702$)[1].

To summarise, deduplication methods based on complete matching of one more key-value attributes from document metadata are prone to generate large number of false positives and they can only identify exact duplicates at best.

In this work, we instead focus on using document similarity measures that are capable of identifying both near and exact duplicates. Further, as detailed in Section 4., our methods will benefit from text processing of document content (both abstract and full text of documents) rather than simple matching of key-value attributes.

3. Deduplication Overview

Deduplication is commonly carried out as three main subtasks: i) indexing ii) comparison and iii) classification. Indexing is the task of identifying key attributes of documents so that documents sharing common value for those attributes can be arranged into subgroups of their own, also called blocks. The comparison step benefits from such groupings since the lookup for duplicates for a given document can be restricted to comparing it with other documents within the same block only. For a given document, the comparison step assigns scores to other documents within its block representing their degree of match. Finally, the classification subtask defines a threshold above which documents having scores are predicted to be duplicates of the input document.

For our deduplication task, we designate **abstracts** of scholarly documents as our key attributes. There are three main reasons that motivate this choice. First, abstracts are an integral part of any scientific document and summarize the central idea being described in the document. Second, abstracts are extensively available (in comparison to full text of documents, for example, which are often limited by copyright issues) and this greatly helps to reduce problem cases with null or missing values. Third, they are easily accessible because repositories usually expose structured bibliographic information e.g. an xml record of scholarly documents including their abstracts. As, we shall observe in Section 5., using document abstracts suffices for achieving a good performance deduplication system.

In Section 4., we describe the details of our deduplication method that integrates two separate methods of identifying duplicates. Each of the individual methods use separate models for the comparison subtask – bitwise matching of hash values and cosine vector similarity of document abstracts, respectively. In the first method, comparison assigns scores ranging from 0 to 64 (the maximum possible bitwise difference in a 64-bit hashing scheme) while cosine similarity value ranges from -1 (completely different) to 1 (exactly same) for the second method. We establish their duplicates classification threshold values empirically (Section 5.2.).

4. Our Approach

4.1. Labeled Dataset Creation

To the best of our knowledge, there are no existing datasets of scholarly documents fit for the purpose of deduplication experiments. We, therefore, build one ourselves and this involves two main tasks. First is the task of obtaining a collection of scholarly documents present across multiple repositories. The second task is then to label each document in this collection with information of duplicates present for each of them within the collection. On completing these tasks, we obtain a labeled dataset of duplicates in a collection of scholarly documents suitable for our deduplication experiments.

For the first task, we use CORE (Knoth and Zdrahal, 2012), the world's largest aggregator of openly accessible scientific documents. At the time of writing this manuscript, CORE consists of more than 177 million of scientific documents aggregated from over $9,867$ repositories around the world. Owing to the issues we highlighted in Section 1., we posit that it contains a significant number of duplicates. We extract $1,687,044$ document records from the CORE such that each document has a title (more than 20 characters long), abstract (more than 500 characters long), full text (more than 5000 characters long) and a DOI conforming to standard regex pattern (Gilmartin, 2015). This helps us in getting started with a collection that is free of missing or null values, unusually short text or incorrect DOIs. Further tasks are needed to improve the quality of our dataset. We convert the title, abstract and full text of documents to lowercase and replace multiple spaces by one. In full text and abstracts, we strip out formatting characters (e.g. newline, tab, space), URLs (using regex pattern), punctuation characters, digits and stop words. In this collection of $1,687,044$ document records, we identify the most frequent $1,500$ sentences (string of text followed by a dot character and a space) and words (token delimited by space) occurring in their full text. Manual analysis shows that the most frequent sentences and words are boilerplate text[2]. Subsequently, we remove any occurrence of these text from all the document abstracts. The full text of documents are no longer needed for our purposes and are dropped.

[1] The frequent titles as well as DOIs are provided as separate files in the dataset we release.

[2] Also included as separate files in the dataset we release.

CORE ID	DOI	Title	Abstract	CORE ID of Duplicates
15080768	10.0000/anziamj.v44i0.707	comparison of time domain ...	analyse centred methodology	[]
93949429	10.1007/JHEP01(2018)055	search for additional heavy neutral ...	neutral bosons prime bosons	[153387874]
29502657	10.1051/0004-6361/201425252	constraining the properties of ...	abridged latest cigale fitting	[52711245, 52427083, 52659917, 52672633]

Table 2: Example entries in our ground truth dataset

Next, we clean our dataset to filter out records with possibly incorrect or malformed DOIs and titles. To address the issue of generic DOIs being assigned to documents, we define a simple heuristic that a DOI (which is defined by prefix/suffix structure) for a document must have a suffix which is not just a sequence of alphanumeric characters only. This helps to filter out documents with DOIs such as 10.1093/bioinformatics (which refers to a journal) but preserve others like 10.1088/0953-8984/21/17/175601. We further remove all those documents in our dataset whose DOI belongs to the list of $1,000$ most frequent DOIs and/or whose title belongs to the list of $8,500$ most frequent titles identified in the CORE's total collection (previously discussed in Section 2.). The resulting dataset amounts to $1,525,199$ document records.

Based on the DOIs, we then proceed to bucket document records into groups such that each group contains all the documents which have the same DOI. A singleton group identifies a non-duplicate document while a group x with n elements; $n > 1$ indicates a duplicate group x with n documents in it having the same DOI. We observe $1,320,551$ non-duplicate groups and $204,648$ duplicate groups. From the non-duplicate groups, we randomly select $50,000$ groups (amounting to an equal number of document records). From the duplicate groups, on one hand, we randomly select $11,473$ buckets (amounting to $25,000$ document records) which meet the criteria that within a bucket all its document records contain exactly matching titles and exactly matching abstracts. On the other hand, we randomly pick $10,448$ buckets (amounting to $25,000$ document records) such that each bucket contains document records whose titles are not all the same and their abstracts differ as well. The former $25K$ records is our approximation of exact duplicates, the latter $25K$ for near duplicates and the first $50K$ for non-duplicates; thereby creating a duplicates/non-duplicates balanced dataset of $100K$ document records – our **Ground Truth** dataset. We release this dataset as a publicly accessible download from `https://core.ac.uk/documentation/dataset/`.

Table 2 shows the schema of our ground truth dataset. For a given document in the dataset, its duplicates are all other documents contained in the same group as that of the input document. The group size of a group is defined as the number of documents present within that group. The rightmost column in Table 2 represents the duplicates (list of CORE IDs) identified for a given document (CORE ID) on the left.

In the resulting collection of duplicates groups, Figure 1 reveals the frequencies of their sizes. Namely, we have $18,083$ different duplicate groups each having 2 duplicates, $2,540$ groups each with 3 duplicates and so on. The duplicate group sizes occurring in the ground truth dataset range from 2 to 14, meaning that duplicate groups are formed by identifying at least 2 documents that are duplicates of each other and in some cases, we observe as many as 14 documents that are duplicates to each other.

4.2. Baseline – Exact Title Matching

Having carefully built the ground truth dataset, we start by establishing a baseline which uses exact title matching method. For an input document x, we retrieve all other documents y in the collection with title string matching exactly that of x. The comparison step assigns a score of 1 to matching documents and 0 otherwise. The set of all y identified for the document x constitutes its duplicates, i.e. the classification is based on the criteria that duplicate documents have a comparison score of 1.

This method, however, has a number of limitations. At best, it can only identify duplicates which exactly match on their titles. Ideally, we would like to have a solution that i) matches exact duplicates ii) is robust to account for near duplicates and iii) provides parameters that can be tuned for specifying desired level of variability in documents' text in order for them to be considered near duplicates. This draws our attention to two distinct methods based on content similarity – the simhash matching method and the document vectors similarity method.

4.3. Simhash Matching

Hash algorithms are functions that map data of arbitrary size (e.g. abstract text of a scholarly document) to data of fixed size (n-bit string). Hashing algorithms have been widely used for document deduplication, as in (Forman et al., 2005), (Hoad and Zobel, 2003), (Bernstein and Zobel, 2004) etc. This is desirable because it allows to implement a uniform approach to comparing variable length documents – documents can instead be compared based on their fixed length hash values. A common choice of a hashing algorithm used for deduplication is the simhash function with $n = 64$ bit encoding scheme. Once the hash values are obtained, documents can be compared based on hamming distance between their hash values. For a given document pair, hamming distance of 0 indicates that the documents are exact duplicates of each other while higher values represent increasing degree of dissimilarity between them. Notably, simhash belongs to the class of locality sensitive hashing functions which have the characteristic property that similar documents, i.e. near duplicates, produce similar hash values, i.e. have low hamming distances. The choice of a particular value of hamming distance is specific to the deduplication task at hand and forms the basis of categorising all documents within that hamming distance as near duplicates for a given input document.

It is evident that simhash fulfills all the requirements we just discussed for our deduplication task. Furthermore, simhash

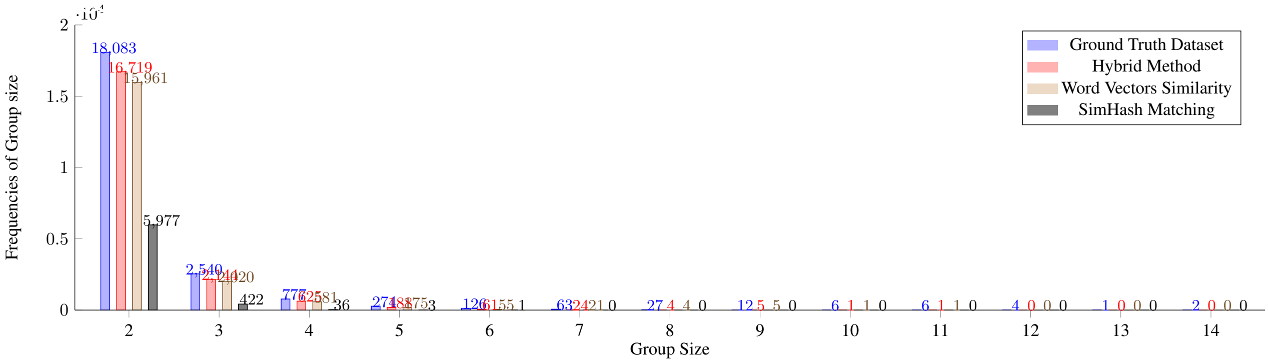


Figure 1: Frequencies of duplicate groups on ground dataset and as predicted by our different methods.

based deduplication has been shown to be a scalable solution for deduplication on document collection of significantly large size (70 million in case of (Sood and Loguinov, 2011) and 8 billion for (Manku et al., 2007)). Naturally, all these factors form the basis of us choosing a simhash based method for finding duplicates. In our implementation, we first map the documents in our collection to their hash values by applying the simhash function to their respective abstracts. This can be obtained as part of pre-processing our document collection and we use an open source implementation of simhash algorithm for the task[3]. Next, the documents are compared based on their hash values. The comparison score assigned to a document is its hamming distance with the input document. Given a threshold value (hamming distance) α, any documents (x,y) can be inferred to be duplicates of each other if they have a comparison score κ such that $\kappa \leq \alpha$.

4.4. Document Vectors Similarity

A widely used approach to text processing is using low dimensional vectors (also known as word embeddings) to represent the meanings of words. The Word2vec algorithm (Mikolov et al., 2013) popularized this approach as a process of training a neural network model to obtain vectors for words based on their distribution in a large text corpus. Since then, many deep learning models have been proposed for the task. BERT (Devlin et al., 2018) is a recently proposed deep learning model for building language representations and it has been shown to produce state-of-the-art results when used for a number of text processing tasks. The final layer of the BERT model outputs n-dimensional[4] vector for each (sub)words in input sentence and these are dense vectors based on the context (i.e. the input sentence). Many pre-trained BERT models are openly available for end user tasks. In this work, we use a pre-trained BERT model released by (Guo et al., 2019) (BERT$_{BASE}$ model trained on BooksCorpus (Zhu et al., 2015) and English Wikipedia text) and use an open source library[5] to obtain word vectors for document text and apply it for our deduplication purposes. Specifically, for each document in our ground truth dataset, we split its abstract text into a list of sentences[6]. We feed the sentences, in turn, to the BERT model and retrieve vectors for each (sub)words as identified by BERT. BERT uses WordPiece tokenization (Wu et al., 2016) to identify (sub)words in the input text and this makes it robust for predicting vectors for out-of-vocabulary words which may occur in our input. We compute a single vector (i.e. a document vector, $\vec{d_x}$) representing a document x as follows:

$$\vec{d_x} = \sum_{m \in Sentences(x)} \frac{1}{|m|} * \sum_{n \in Words(m)} \frac{\overrightarrow{BERT(n,m)}}{|n|}$$

where $\overrightarrow{BERT(n,m)}$ is the vector identified by BERT model for word n in sentence m.

We compute document vectors for each of the documents in our ground truth dataset and use that as a basis of determining similarity of the documents. For any document pair (y,z), the comparison score κ is the cosine similarity value of their document vectors($\vec{d_y}, \vec{d_z}$) and the documents are considered to be duplicates of each other if $\kappa \geq \beta$ for some classification threshold β.

4.5. Hybrid Method

The simhash matching method and the document vector similarity method inherently work on a different level of textual representation. The former method treats document abstracts at a structural level – looking for overlap of words or characters in surface representation of text. The document vector method, on the other hand, approaches deduplication from the perspective of meaning similarity.

We therefore propose a hybrid method to deduplication which makes use of both these methods. Our motivation here is to understand whether these methods complement each other in building a better deduplication system. Given different thresholds for simhash similarity ($\alpha\_1, \alpha\_2 \ldots\ldots \alpha\_m$) and document vector similarity ($\beta\_1, \beta\_2 \ldots\ldots \beta\_n$) methods, the hybrid method generates prediction for duplicates as outlined in Algorithm 1.

[3] `https://github.com/seomoz/simhash-py`

[4] 768 (BASE model) or 1024 (LARGE model)

[5] `https://github.com/imgarylai/bert-embedding`

[6] We use a simple regular expression that splits on full stop and question marks.

Algorithm 1 Hybrid Method

```
function INTEGRATION(alpha,beta)
    results = { }
    for doc in ground_truth_dataset do
        set_x = SIMHASH_MATCH(doc, ground_truth_dataset, alpha)
        set_y = DOCVEC_MATCH(doc, ground_truth_dataset, beta)
        set_z = UNION (set_x, set_y)
        results[doc_x] = set_z
    end for
    return results
end function
simhash_alphas=SET($\alpha\_1, \alpha\_2 \ldots\ldots \alpha\_m$)
docvec_betas=SET($\beta\_1, \beta\_2 \ldots\ldots \beta\_n$)
hybrid_results = { }
for x in simhash_alphas do
    for y in docvec_betas do
        hybrid_results[(x,y)] = INTREGATION(x,y)
    end for
end for
```

5. Experiments and Evaluation

5.1. Evaluation Metrics

To evaluate our methods, we use the standard metrics of precision and recall for both duplicate and non-duplicate classes. In addition, we report the macro F1 average and accuracy values to reflect overall performance.

As we observed in Table 2, for any given document (say d), there can be a set (say X_d) consisting of zero or more documents in the ground truth dataset labelled as its duplicates. Likewise, for the same input document d, predictions from our methods can result in a set (say Y_d) of documents as duplicates. Under each of our experiments, we can identify a prediction Y_d to belong to one of the following categories.

- a **true positive (TP)** if $X_d \subset Y_d \ and \ X_d \neq \phi \ and \ Y_d \neq \phi$
- a **false positive (FP)** if $Y_d \neq \phi$, and $(X_d \not\subset Y_d \ or \ X_d = \phi)$
- a **true negative (TN)** if $X_d = Y_d = \phi$.
- a **false negative (FN)** if $Y_d = \phi$ but $X_d \neq \phi$

The confusion matrix looks like the one shown in Table 3.

	$X_d \neq \phi$	$X_d = \phi$
$Y_d \neq \phi$	$(X_d \subset Y_d) \implies$ **TP** $(X_d \not\subset Y_d) \implies$ **FP**	**FP**
$Y_d = \phi$	**FN**	**TN**

Table 3: Confusion matrix

By conducting our experiments over all the documents present in the ground truth dataset and noting their predictions, we can compute the count of true positives, false positives, true negatives and false negatives. Based on the confusion matrix, we evaluate the outcome of an experiment using the standard metrics of precision, recall, accuracy and macro-F1.

5.2. Experiments

The baseline method is non-parametric and does not have multiple classification thresholds to define. Therefore a single iteration of this method on the ground dataset is sufficient to understand its performances.

The simhash matching method is defined by a number of parameters. These include:

- **Content Unit**: Simhash is based on the principle of building representation of total content by composing representations obtained for smaller units of data that make the content. Typically, text can be represented as sequence of characters or words and we explore the possibility of using both these content units in implementing our simhash method.
- **Shingles Size**: The hash values obtained by simhash are characterized by the span of content units that involve in making a single unit of representation. Often referred to as 'shingles', we can specify the value for it's size to define what number number of content units (words/characters) in sequence should be considered as a single token while building up the document representation.
- **Hamming Distance**: Hamming distance is the number of positions at which bits of two hash values differ. By specifying different values of hamming distance, we can define different thresholds for the deduplication classification subtask. Document pairs under consideration are considered to be duplicates if their hamming distance does not exceed the threshold value.

We experimented with 338 different configurations of this method and obtained different evaluation scores. Figure 2a and 2b show the different macro F1 scores obtained for different choices of shingle size and hamming distance when using words and characters as content unit, respectively.

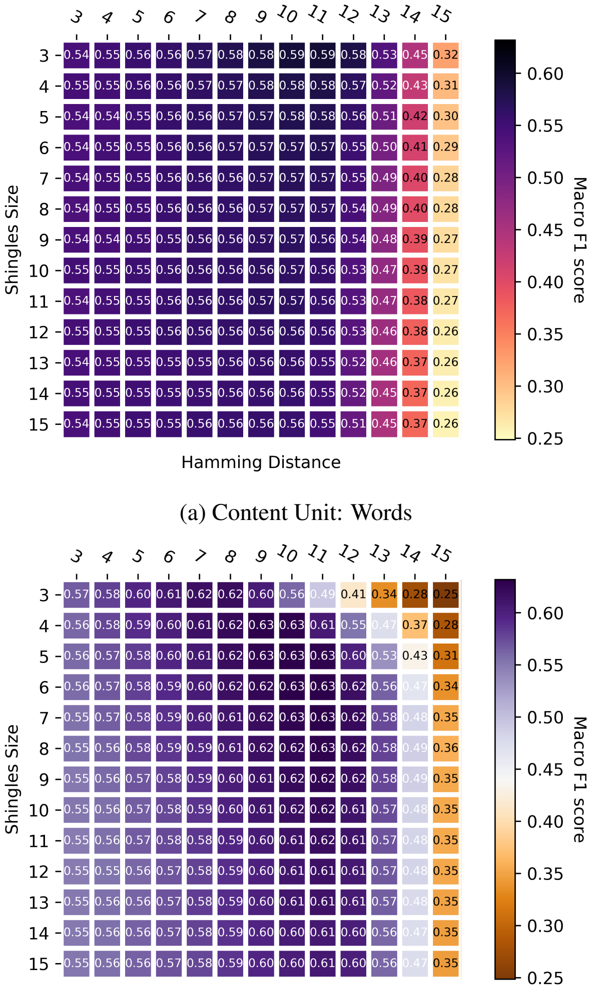
For the document vector similarity method, a number of different prediction scores can be obtained by using different threshold values for classification. Figure 3 shows the macro F1 scores obtained on using different choices of cosine similarity values as the threshold value for duplicate classification. In total, we experimented with 19 different threshold values.

The hybrid method benefits from multiple combination possibilities of parameter values of the simhash and document vectors similarity methods. In total, we evaluate the hybrid method on $6,422$ unique configuration of parameter values resulting from such combinations.

In Table 4, we list the best scoring configuration of the parameters for each of our methods and Figure 1 shows the group size frequencies of the duplicates they predict. Overall, we see that the hybrid method has the best scoring macro F1 score and has a similar distribution of group size frequencies as observed for the ground truth dataset. This indicates that the hybrid method is the best for predicting both duplicates and non duplicates and is, therefore, the deduplication model of our choice.

Method	Parameters				Precision Duplicates	Recall Duplicates	Precision Non Duplicates	Recall Non Duplicates	Macro F1 score	Accuracy
	Shingling Content	Shingles Size	Hamming Distance	Cosine Similarity						
Exact Title Matching	NA	NA	NA	NA	0.830	0.50	0.709	0.992	0.757	0.746
Simhash Matching	Character	5	10	NA	0.697	0.247	0.598	0.985	0.631	0.616
Document Vector Similarity	NA	NA	NA	0.98	0.912	0.779	0.861	0.986	0.885	0.883
Hybrid Method	Character	5	10	0.98	0.908	0.828	0.899	0.979	0.904	0.903

Table 4: Evaluation scores obtained for best performing configuration of different methods



(a) Content Unit: Words

(b) Content Unit: Characters

Figure 2: HeatMaps showing Macro F1 scores for simhash matching using different parameter values

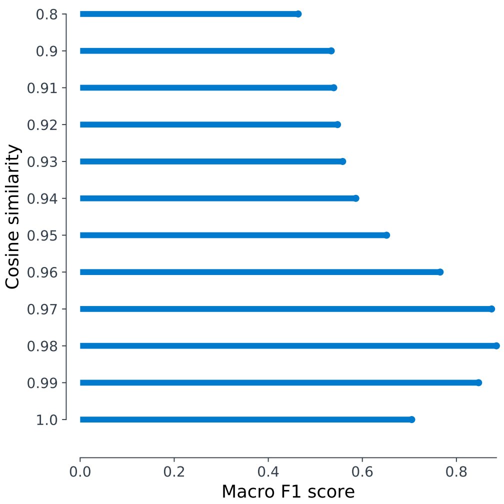


Figure 3: Macro F1 scores obtained for different Document vector similarity approach

6. Results & Discussion

Looking into the evaluation scores in Table 4, the simhash matching method by itself does not seem to perform any better than the baseline method. However, for reasons discussed earlier, exact matching approaches would lead to a large number of false positives in a real world scenario; especially with matching titles. Our ground truth dataset was carefully curated to avoid erroneous titles and therefore the evaluation scores can be expected to be in favor of the baseline method. What our experiment demonstrates instead is that simhash matching method can be a good starting ground for identifying duplicates with variations in their content. Further, we note that the document vector similarity method is the main contributor towards obtaining significant performance gains. This likely signifies that duplicates of scholarly documents are not simply variations in character/string positions but are rather semantically related paraphrases of content. Allowing for both the structural variation and meaning representation of text, we observe that the hybrid model achieves the best performance score. Apart from the perspective of gaining better evaluation scores, there are also pragmatic reasons to adopt a hybrid method. This is evident in a real world deduplication scenario since we host our deduplication service as an openly accessible web API at `https://core.ac.uk/docs/#!/articles/nearDuplicateArticles`. We notice that the users would like to (optionally) obtain duplicates based on one or more pieces of additional information they may have (e.g. author names + title + year of publication) rather than specifically looking for duplicates based on similarity of abstract text only. In such cases, we take a step further and integrate (in much the same manner as done for the hybrid method) the results obtained by exact matching of user supplied attributes with the results obtained from our hybrid method for serving API responses.

In Figure 1, we see a long-tailed distribution of duplicate group sizes with very low frequencies in the ground truth dataset. On manual examination, we notice that some of these low-frequency groups are formed because of incorrectly assigned DOIs. Despite taking great care in filtering documents with erroneous DOIs, we are not able to automatically filter out all such DOIs in our ground truth dataset. The incorrect DOIs can lead to fewer number of duplicates identified for a group during the ground truth dataset creation. This can result in our methods (which are based on comparing similarities of abstract text) predicting higher number of duplicates for an input document than those identified for it in the ground truth dataset. For this reason, we considered a prediction to be true positive (in Section 5.1.) if it contained all the elements of the labelled set and not necessarily both these being equal. The incorrect DOIs and/or other erroneous metadata information do, in fact, propagate from source repositories where they are originally hosted and there remains very little at our end to try and resolve these issues.

In this work, we only considered documents with English text. Many different factors motivated this choice; mainly the ubiquitous support for English language text processing; availability of open source libraries and pre-trained word embedding models on large corpus of English text. Many other pre-trained word embeddings (e.g. (Bojanowski et al., 2016), (Beltagy et al., 2019)) are also available apart from the one we used in this work. Further experiments will be needed to study the performance of our method under these settings.

7. Related Work

A number of previous studies have presented deduplication in the context of a variety of practical applications. Examples include deduplication for detecting plagiarised content (Hoad and Zobel, 2003), (Bernstein and Zobel, 2004), improving quality of web search (Manku et al., 2007), (Su et al., 2010), (Syed Mudhasir et al., 2011), finding similar files in document repositories (Manber, 1994), (Forman et al., 2005), measuring source code similarity of software systems (Yamamoto et al., 2005). Broadly speaking, existing work on deduplication can be classified into two main categories based on the approaches they adopt. In the first category of work, we see deduplication approach based on matching of values of attributes that make up the data items. This approach is fairly common with deduplication of records present in structured content systems such as databases (e.g. (Chaudhuri et al., 2003)). The second category of approaches are based on comparing semantic similarity of document contents. For example, (Forman et al., 2005), (Manber, 1994), (Shenoy et al., 2017) use different hashing functions (MD5 hash, minhash etc.) over document text to obtain document hash values. Likewise, (Bogdanova et al., 2015), (Zhang et al., 2017) use Word2vec (Mikolov et al., 2013) embeddings to represent questions posted in online user forums and use that for identifying semantically related question pairs. Training machine learning models (Su et al., 2010) and more recently, deep learning (Mudgal et al., 2018) have also been proposed in this regard. In comparison, our work uses i) simhash function; a locality sensitive hashing function introduced by (Charikar, 2002) ii) word embeddings coming from the BERT model (Devlin et al., 2018) and iii) builds upon the power of pre-trained language representation model instead of training a neural network specific to the purpose.

More related to our study are works focusing on deduplication of scholarly data. (Jiang et al., 2014) define a multi-step rule-based method for deduplication of bibliographic metadata records (BibTeX records) of biomedical scholarly documents. They use exact matching on attribute-value pairs (e.g. DOI, repository specific identifier such as the PubMed ID number, author names) of the records; (Qi et al., 2013) also put manual effort to correctly identify duplicates on such databases. (Canalle et al., 2017) define several metrics (repetition, distinctiveness, density etc.) to study the importance of different attributes of bibliographic datasets when used for deduplication task. A recent work (Atzori et al., 2018) studies deduplication of entities related to scholarly publication (e.g. datasets, organizations, research funders) as present in big scholarly communication graphs such as the OpenAIRE scholarly communication graph (`https://api.openaire.eu/`). In terms of content based approaches to scholarly document deduplication, (Labbé and Labbé, 2013) study forgeries of research outputs published in a few conferences and use inter-textual distance as a measure of document similarity. The authors define their own measure of inter-textual distance based on word frequencies but it is not clear how it would compare to other highly successful methods which have been reported for deduplication of documents outside the scholarly domain text. For example, locality sensitive hashing method has been successfully used for deduplication of web corpus (Manku et al., 2007), technical documentations (Forman et al., 2005) and clinical notes (Shenoy et al., 2017). Similarly, word vectors have been used for deduplication of related question pairs (Bogdanova et al., 2015). In our work, we pursue the study of deduplication of **scholarly documents**. Like (Labbé and Labbé, 2013), we follow the content based approach to deduplication but build upon the strength of both locality sensitive hashing and word embeddings methods. These methods were studied in isolation for specific data collections in the past but our work shows that both these methods produce results which complement each other and therefore, a hybrid method should be used for obtaining the best performing model for deduplication of scholarly documents.

8. Conclusion

We produced a labelled dataset of $100K$ scholarly documents suitable for deduplication studies and proposed a novel method to deduplication of scholarly documents – a hybrid method using simhash and document vectors similarity. With an extensive set of experiments, we established the optimal values for the parameters of the hybrid method; achieving a macro F1-score of 0.90 and an accuracy of 90.30%. This is well above the performance obtained from a baseline system and over the individual methods making up the hybrid method. As a practical outcome of our research, we deploy our deduplication service as a publicly accessible web API and publicly release our dataset to the global audience.

9. Bibliographical References

Atzori, C., Manghi, P., and Bardi, A. (2018). Gdup: De-duplication of scholarly communication big graphs. In *5th IEEE/ACM International Conference on Big Data Computing Applications and Technologies, BDCAT 2018, Zurich, Switzerland, December 17-20, 2018*, pages 142–151.

Beltagy, I., Cohan, A., and Lo, K. (2019). Scibert: Pretrained contextualized embeddings for scientific text.

Bernstein, Y. and Zobel, J. (2004). A scalable system for identifying co-derivative documents. In Alberto Apostolico et al., editors, *String Processing and Information Retrieval*, pages 55–67, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bogdanova, D., dos Santos, C., Barbosa, L., and Zadrozny, B. (2015). Detecting semantically equivalent questions in online user forums. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 123–131.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Canalle, G. K., Lóscio, B. F., and Salgado, A. C. (2017). A strategy for selecting relevant attributes for entity resolution in data integration systems. In *Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, pages 80–88. INSTICC, SciTePress.

Charikar, M. S. (2002). Similarity estimation techniques from rounding algorithms. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 380–388, New York, NY, USA. ACM.

Chaudhuri, S., Ganjam, K., Ganti, V., and Motwani, R. (2003). Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 313–324. ACM.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Forman, G., Eshghi, K., and Chiocchetti, S. (2005). Finding similar files in large document repositories. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 394–400, New York, NY, USA. ACM.

Gilmartin, A. (2015). Dois and matching regular expressions. `https://www.crossref.org/blog/dois-and-matching-regular-expressions/`.

Guo, J., He, H., He, T., Lausen, L., Li, M., Lin, H., Shi, X., Wang, C., Xie, J., Zha, S., Zhang, A., Zhang, H., Zhang, Z., Zhang, Z., and Zheng, S. (2019). Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing.

Hoad, T. C. and Zobel, J. (2003). Methods for identifying versioned and plagiarized documents. *J. Am. Soc. Inf. Sci. Technol.*, 54(3):203–215, February.

Jiang, Y., Lin, C., Meng, W., Yu, C., Cohen, A. M., and Smalheiser, N. R. (2014). Rule-based deduplication of article records from bibliographic databases. *Database*, 2014:bat086.

Klein, M., Broadwell, P., Farb, S. E., and Grappone, T. (2016). Comparing published scientific journal articles to their pre-print versions. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, JCDL '16, pages 153–162, New York, NY, USA. ACM.

Knoth, P. and Zdrahal, Z. (2012). Core: three access levels to underpin open access. *D-Lib Magazine*, 18(11/12).

Kołcz, A., Chowdhury, A., and Alspector, J. (2003). Data duplication: An imbalance problem?

Labbé, C. and Labbé, D. (2013). Duplicate and fake publications in the scientific literature: how many scigen papers in computer science? *Scientometrics*, 94(1):379–396.

Manber, U. (1994). Finding similar files in a large file system. In *USENIX WINTER 1994 TECHNICAL CONFERENCE*, pages 1–10.

Manku, G. S., Jain, A., and Das Sarma, A. (2007). Detecting near-duplicates for web crawling. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 141–150, New York, NY, USA. ACM.

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

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., Krishnan, G., Deep, R., Arcaute, E., and Raghavendra, V. (2018). Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 19–34, New York, NY, USA. ACM.

Qi, X., Yang, M., Ren, W., Jia, J., Wang, J., Han, G., and Fan, D. (2013). Find duplicates among the pubmed, embase, and cochrane library databases in systematic review. *PLoS One*, 8(8):e71838.

Shenoy, S., Kuo, T.-T., Gabriel, R., McAuley, J., and Hsu, C.-N. (2017). Deduplication in a massive clinical note dataset. *arXiv preprint arXiv:1704.05617*.

Sood, S. and Loguinov, D. (2011). Probabilistic near-duplicate detection using simhash. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1117–1126, New York, NY, USA. ACM.

Su, W., Wang, J., and Lochovsky, F. H. (2010). Record matching over query results from multiple web databases. *IEEE transactions on Knowledge and Data Engineering*, 22(4):578–589.

Syed Mudhasir, Y., Deepika, J., Sendhilkumar, S., and Mahalakshmi, G. (2011). Near-duplicates detection and elimination based on web provenance for effective web search. *International Journal on Internet & Distributed Computing Systems*, 1(1).

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation.

Yamamoto, T., Matsushita, M., Kamiya, T., and Inoue, K. (2005). Measuring similarity of large software systems based on source code correspondence. In Frank Bomarius et al., editors, *Product Focused Software Process Improvement*, pages 530–544, Berlin, Heidelberg. Springer Berlin Heidelberg.

Zhang, W. E., Sheng, Q. Z., Lau, J. H., and Abebe, E. (2017). Detecting duplicate posts in programming qa communities via latent semantics and association rules. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1221–1229. International World Wide Web Conferences Steering Committee.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.