Finding Translations in Scanned Book Collections

Ismet Zeki Yalniz
Dept. of Computer Science
University of Massachusetts
Amherst, MA, 01003
zeki@cs.umass.edu

R. Manmatha
Dept. of Computer Science
University of Massachusetts
Amherst, MA, 01003
manmatha@cs.umass.edu

ABSTRACT

This paper describes an approach for identifying translations of books in large scanned book collections with OCR errors. The method is based on the idea that although individual sentences do not necessarily preserve the word order when translated, a book must preserve the linear progression of ideas for it to be a valid translation. Consider two books in two different languages, say English and German. The English book in the collection is represented by the sequence of words (in the order they appear in the text) which appear only once in the book. Similarly, the book in German is represented by its sequence of words which appear only once. An English-German dictionary is used to transform the word sequence of the English book into German by translating individual words in place. It is not necessary to translate all the words and this method works even with small dictionaries. Both sequences are now in German and can, therefore, be aligned using a Longest Common Subsequence (LCS) algorithm. We describe two scoring functions TRANS-cs and TRANS-its which account for both the LCS length and the lengths of the original word sequences. Experiments demonstrate that TRANS-its is particularly successful in finding translations of books and outperforms several baselines including metadata search based on matching titles and authors. Experiments performed on a Europarl parallel corpus for four language pairs, English-Finnish, English-French, English-German, English-Spanish, and a scanned book collection of 50K English-German books show that the proposed method retrieves translations of books with an average MAP score of 1.0 and a speed of 10K book pair comparisons per second on a single core.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.7 [**Digital Libraries**]: Collection, Systems Issues

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.

General Terms

Algorithms, Experimentation

Keywords

Translation detection, sequence alignment, unique words, book collections

1. INTRODUCTION

This paper describes an approach to finding translations of documents which are long and noisy - specifically scanned books with OCR errors in large collections such as the Internet Archive (IA) or Google Books. However, it is also applicable to documents produced by governments and companies.

Finding translations is useful for many reasons. It will enable search engines to display translated versions of a book as part of the results so that for example a Spanish reader may choose a Spanish version of Goethe's Faust. By finding translations one can create parallel corpora for creating better machine translation algorithms and for cross-lingual search systems. The humanities and library communities have a great interest in aggregating works and finding translated versions of books such as Goethe's Faust. IFLA's Functional Requirements for Bibliographic Records (FRBR) requires that the next generation of cataloging systems include works aggregation [22]. This will include information on which books are translated versions of each other. However, no specific technique is proposed to do the FRBR-ization and it is implicitly assumed that metadata will be sufficient. Experiments show that metadata is not accurate enough to always determine which books are translations.

There are two distinct problems in the context. The first problem, which is the focus of this paper, is to decide whether given two books are translations of each other and to do it for all book pairs in the collection. Given that most book pairs are not translations, comparing all book pairs can be expensive since there are $O(nm)$ distinct book pairs in collection of n books in one language and m books in the other. Hence, there is a need for an efficient approach. The second problem is to map the portions of translated text between any two books in different languages. This is not the focus of this paper although we provide Figure 1 to illustrate translated portions of two example books.

Books and translations of books have many interesting characteristics. Books are usually much longer than web documents. Texts obtained from scanned books have also character recognition errors - in some cases substantial - and

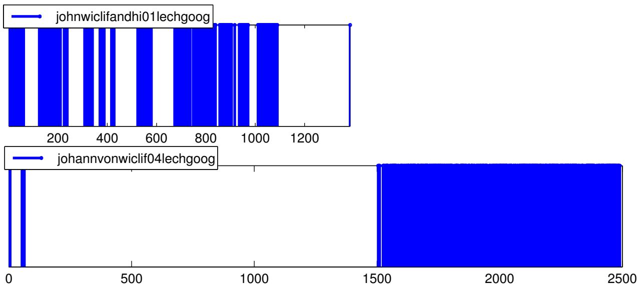

Figure 1: The figure shows the approximate overlap between an English translation (upper bar) and the German original (lower bar) as determined by a global alignment algorithm. The lengths of the bars reflect the relative sizes of the two books. Blue (black) denotes aligned portions. The German version contains the complete text while the English version is only Volume II and hence the big gap in the lower bar. The English version has additional notes and these are reflected by the gaps in the upper bar.

any algorithm must cope with them. Most translations do not have one-to-one overlap. Figure 1 shows the automatically generated overlap between Wiclif's biography in the original German and a translated version in English which only includes volume 2 with additional notes[1]. The figure shows that only a portion of the two texts overlap.

One approach is to use the book metadata to find translations of books. Our experience, however, is that metadata entries can be erroneous and therefore they are not completely reliable. This approach, therefore, does not solve the problem as discussed further in the experiments section. There are several types of errors in the metadata of scanned books. First of all, the language of books are often specified incorrect. In a test collection of 378 books, the language of several books was incorrectly specified - they are marked as English even though they are clearly in German or vice versa. Books written in multiple languages are typically not clarified too. There are books marked as English although they are in German with an English preface and/or notes. Even if the metadata is correct, it is sometimes not easy to tell whether two books are translations or not. Quite often titles do not translate exactly to other languages. Even though two books have the same title after translation, the translated version may have only the translator's or editor's name as the author. Metadata entries are manually entered to the system by the people who scan books, therefore the process is error prone. A similar problem does also exist for different Wikipedia articles. While some articles are direct translations of each other, many articles with the same title are actually written by different authors and therefore they are not translations. Therefore, Wikipedia articles can not be used for building translation detection corpora since the ground truth is not clear.

Techniques have been previously suggested for finding near duplicates in the same language using shingling (n-gram overlap) [4, 5] or even partial duplicates using the alignment of "unique words" [30]. The applicability of such techniques to translation detection is not trivial. Word order is not usually preserved across languages and hence translations of individual words in a book using a dictionary do not preserve n-grams of words. Thus, traditional shingling techniques are not directly applicable for translation detection. In addition most free dictionaries available online are small. For example, the largest English-German dictionary available to us has 62K entries while a desktop edition of Merrian Webster's Collegiate dictionary has 225K entries. Due to the fact that morphological variants of words are often not found in small size dictionaries, less words get translated. Another option is to use a machine translation system to translate all the books to a common language and apply mono-lingual duplicate detection techniques as Uszkoreit et al. [29] used at Google. However, this approach requires building robust translation systems for each language and the actual translation stage is computationally expensive. Given that most researchers and organizations do not have Google's computational resources, a more practical solution is needed. Krstovski and Smith [19] use hapax words, i.e., words which are common between two different languages, to identify translation pairs in scanned book collections. They adopt a vector space representation for books and use Cosine distance as the translational similarity metric. The weakness of this approach is that there is no guarantee there exists

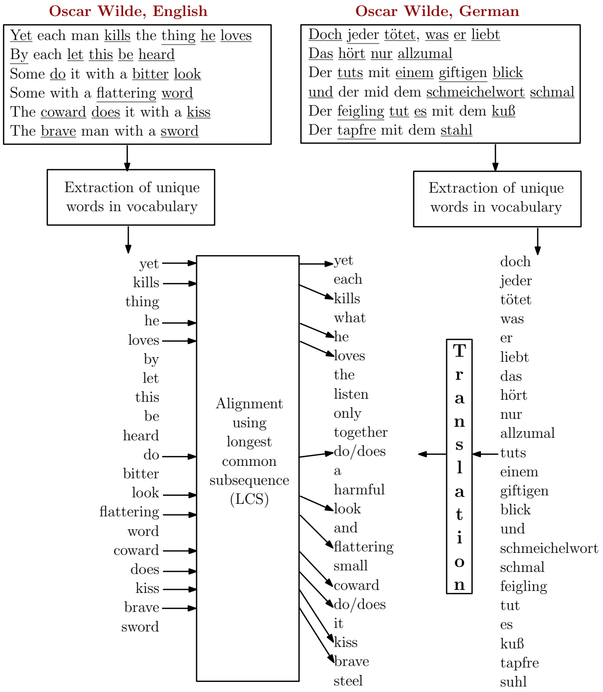
Oscar Wilde, English
Yet each man kills the thing he loves
By each let this be heard
Some do it with a bitter look
Some with a flattering word
The coward does it with a kiss
The brave man with a sword
Oscar Wilde, German
Doch jeder tötet, was er liebt
Das hört nur allzumal
Der tuts mit einem giftigen blick
und der mid dem schmeichelwort schmal
Der feigling tut es mit dem kuß
Der tapfre mit dem stahl
Extraction of unique words in vocabulary
Extraction of unique words in vocabulary
yet
kills
thing
he
loves
by
let
this
be
heard
do
bitter
look
flattering
word
coward
does
kiss
brave
sword
Alignment using longest common subsequence (LCS)
yet
each
kills
what
he
loves
the
listen
only
together
do/does
a
harmful
look
and
flattering
small
coward
do/does
it
kiss
brave
steel
Translation
doch
jeder
tötet
was
er
liebt
das
hört
nur
allzumal
tuts
einem
giftigen
blick
und
schmeichelwort
schmal
feigling
tut
es
kuß
tapfre
suhl

Figure 2: Illustration of the proposed framework. Unique words are underlined for two versions of a poem by Oscar Wilde. Unique words from the German version are first translated in to English using a dictionary. The resulting word sequence is aligned with the unique words extracted from the English version using LCS. The words in the LCS are indicated with single headed arrows. It is seen that a large number of words follow the same order in both sequences. This is a clear indication for texts being translations.

[1] The figure is generated as follows: the words which appear more than 20 times in the entire text are filtered out in both books. The remaining words in the English book are translated in place to German using a word dictionary and aligned with the remaining words in the German book using LCS. For visualization purposes we use a binning approach where each bin in the figure is colored blue (black) if there are more than a specified amount of matching words in the range. The bin size is 100 words and the horizontal axis shows the number of bins for each book.

hapax words between all pairs books. Their results also indicate that their approach fail for languages with different language families, such as English and Arabic.

To detect translations we exploit the fact that a translation must preserve the long range order of events and/or ideas. That is, chapter 5 must precede chapter 6 in both English and German versions of "The Lord of the Rings" even though individual sentences (and even paragraphs) do not preserve the word order across languages. Inspired by the work on mono-lingual partial duplicate detection of [30], we show that the sequence of words which occur only once in a book is sufficient to identify translations of books. Consider two books in two different languages, say English and German. The first step is to extract the sequence of words which occur only once in both books. Those words are referred as *unique words*. An English-German dictionary is used to transform the word sequence of the English book into German by translating individual words in place. Many words may end up being not translated since they do not exist in the dictionary. Some words may have multiple translations which are all included in the translated sequence. It turns out that a small fraction of the words being translated is sufficient for our purposes. Hapax words which are common in both sequences (examples of such words may include names which are not translated) are also included in the translated sequence. The resulting sequence is now in German and therefore can be compared with other German books. Comparison is performed using global alignment, specifically Longest Common Subsequences (LCS) algorithm. The length of LCS is a clear indication of translations. Two scoring functions are proposed: TRANS-cs and TRANS-its which normalize the LCS length by the length of the sequences in different ways. See Figure 2 for an illustrative example of our methodology.

Experiments performed on non-noisy EUROPARL documents for several languages and collections of real scanned book collections demonstrate that TRANS-its is very effective and fast in identifying translations. Three different evaluation measures are defined and very high performance scores are obtained for four language pairs of the EUROPARL dataset. English-Finnish experiments show that the technique works across language families. The technique also works on the noisy OCR output of scanned books as well. On a scanned book corpus of 2K English-German books, precision and recall score of 1.0 are achieved (outperforms Krstovski and Smith's method [19]). Retrieval experiments including a scanned book collection of size 50K indicate that TRANS-its achieves a MAP of 1.0. We compare our results to several baselines including metadata search and show that TRAN-its outperforms the baselines over all evaluation metrics. The proposed method is also quite scalable. With simple optimizations, it is seen that TRANS-its compares 10K books per second on a single core.

In the next section, we discuss the related work on translation identification and also provide a brief discussion on mono-lingual duplicate detection methods. Section 3 explains the proposed translation identification framework along with the unique word representation and the scoring functions. Evaluation measures, datasets and experiments are described next. Finally, conclusions are given along with future research directions.

2. RELATED WORK

The related problem of near duplicate detection in the same language has been well discussed especially for web documents. Most of the work uses either fingerprinting algorithms or relative frequency techniques (words with similar frequencies) [4]. Fingerprint techniques [4, 5] divide a document into distinctive chunks or shingles. The standard approach is to use n-grams of words or characters and subsample them using a variety of sampling techniques [14]. Relative frequency techniques assume that two documents with similar words and frequencies must be similar or duplicated [14, 27]. We note that n-grams are not well preserved across languages since word order in a sentence can change across translations. [30] find partial duplicates in collections of books by finding sequences of unique words and then aligning these sequences of unique words. However, their work is restricted to books in the same language. Our work is inspired by their approach.

There has been work on finding comparable corpora for machine translation. Much of this work has been done on either finding parallel sentences from small corpora [28] or web pages [23, 26, 28, 32]. Most of the work on finding web page has utilized structural information - HTML markup such anchors, links, filenames - to find [23, 26] parallel resources. Alignment was specifically rejected as being too expensive. [32] limited the alignment to titles and a translation dictionary to find parallel texts. Much of the machine translation work seems to be on the extraction of bilingual dictionaries [11] rather than finding document translations in large corpora. [28] is one of the few papers on identifying translations. The paper used several translation dictionaries and then computed the word overlap. Filtering was done based on document length for efficiency. The method was tested on a small dataset of about 1000 sentence pairs and another dataset of 325 web document pairs. [25] combined structural and content features to mine web pages for parallel corpora. [21] also used structural features paired with a content filtering scheme to find parallel corpora on the web. [18] used the idea that similar texts would have similar graph structures after compression to find translations of portions of texts.

Uszkoreit et al. [29] is one of two papers to find translations of books. They use Google's large computing resources to translate all the books in the collection to English. This transforms the problem of finding translations to monolingual duplicate detection. Next, they match chunks (n-grams) of words in translated texts to determine translation pairs. One drawback of this approach is that it requires building machine translation systems for all languages and translation of books is computationally expensive. Ideally, one should be able to find translations of books without having to translate them explicitly. The success of their approach is evaluated partially on a small dataset. Uszkoreit et al.'s method is further discussed in the experiments section. Krstovski and Smith [19] use words which are common between translations of books to find translations of books. Each book is represented in the vector space and the translational similarities between books are defined by several distance measures such as Cosine distance. They use Locality Sensitive Hashing (LSH) to efficiently compute the translational similarity scores. Our technique is compared to their approach on the publicly available datasets and we demonstrate that our approach is more accurate.

There has been extensive work on mono-lingual and cross-lingual plagiarism detection. Global alignment methods have been used to find plagiarized passages in the same language [7] but it is impractical for long documents and large collections. Most plagiarism detection techniques instead use a prefiltering stage which involves chunk overlap to detect possible duplicates before the global alignment [9]. Sequence alignment, word sampling and variants of chunking methods have also been tried for cross-lingual plagiarism detection. Please refer to [24] for a recent survey of those methods. It should be noted that cross-lingual plagiarism and translation detection for scanned book collections are different problem domains. Scanned book collections include very long documents with severe amount of OCR errors which prohibit the use of conventional approaches.

3. OUR FRAMEWORK

The first stage of our framework is to identify the language of each book in the collection. This stage can be removed in case the languages of books is known reliably. The second stage involves extracting unique word sequences from all the books. This process is performed once for each book in the collection. In the final stage, all the book pairs between the source and target languages are aligned using Longest Common Subsequences. A translation score is calculated for each book pair based on the length of the LCS. This score is later used for classification and ranking of translation pairs. The details of each stage are elaborated in the following subsections.

3.1 Language Identification

Translation identification require that the language of the book be known. One approach to detect the language is to use the metadata, which is not always reliable. Language identification has been done in the past using stopwords and letter bigrams/trigrams. While letter bigrams/trigrams tend to be more accurate for short passages, on longer texts stopword counts work equally well [12]. Here we use the stopword approach to determine the language of the book. Stopwords for each language (English, French, German, Greek, Italian, Latin and Spanish) are learned from 20 noise free e-books downloaded from the Gutenberg archive. The top five most frequent stopwords are used. A stopword is appropriate for language identification as long as it is not a stopword in another language. This approach makes the language identification process simple, fast and easily generalizable for other languages. A more accurate check on OCR errors can be done using a dictionary but this would be slower and more expensive to create. Note that this technique may fail if the book has high rates of OCR errors which corrupts a large proportion of stopwords. A quick check on a mix of 378 English-German books reveals an accuracy of 100%.

3.2 Extraction of Unique Words

Each book in the collection is represented by the sequence of words which appear only once in the entire text of the book. In this context these words are referred as “unique words”. This sequence of unique words is highly descriptive of the content and flow of ideas in the book. This representation is quite compact. There are are typically a few thousands of unique words for a book of size 100K words. The number of unique words increase as the amount of document noise and the length of the text increases. In a non-noisy book, every second sentence of the document is expected to contain a unique word. The unique word representation is highly tolerant to OCR errors for duplicate and translation detection purposes.

Punctuation and numeric characters are ignored at all stages. This also eliminates false matches caused by matching page numbers which by themselves form a consistent sequence between any two books. Hyphenated words are quite common at the end of each line and they are also corrected automatically before proceeding. For efficiency, unique words are precomputed and stored in binary files. Each unique word is represented by a 32-bit hashcode which is generated using a product sum algorithm over the entire text of the string. For batch processing, the sequences of hashcodes are appended one after another in to binary files which are referred to as “barrels”. A barrel containing 2K books occupies 25-35 megabytes of disk space. Alternatively, one could also index unique words and assign a term ID for each unique word. However, it would be a two-pass approach with large memory and computation requirements since the vocabulary of scanned book collections becomes arbitrarily large as the size of the collection grows.

It should be noted that a unique word in one book may not be necessarily unique in another print version of the same book. This happens due to OCR errors and/or additional or missing text in the other book. Despite these factors, it is still highly probable to find a large number of common words between the two sequences preserving the same order for mono-lingual books. Here we show that this representation is also sufficient to find translation pairs at the book level.

3.3 Translation of Word Sequences

Consider a pair of books - for example one in English and the other in German. At this point we have two unique word sequences extracted from these two books. The aim is to map the unique word sequence from the English book to German or vice versa. The first stage of mapping is to include the common words across translations (names are sometimes preserved across languages) in the translated sequence. For the remaining words, we use a dictionary to translate them in place to German word by word. If there are multiple translations for a word, then they are also included in the translated sequence. It is clear that the translated word sequence may include words repeated more than once after translation, but this is not an issue for the technique.

3.3.1 Preserving common words across translations

Names of people and places are sometimes the same in both texts (i.e. not translated). They have high discriminatory power and it is desirable to incorporate them in to the analysis. For this purpose we first intersect and find all common unique words prior to any translation. Then, the list of common words is interleaved with the translated unique word sequence and sorted based on their original location in text. Notice that names and places may be changed in the translated version of the book. In that case, we still have the translations of the unique words in the sequence which are sufficient to identify translation pairs.

3.3.2 Translation of unique words

The translation lexicon is an important component of the translation identification framework. Larger dictionaries help translate more unique words since they are more likely to be found. It is desirable that the translation lexicon has as many inflections and forms of the word as possible for best performance - since we do not do any morphological processing. Our alignment algorithm (described later) will only match two words if they have the same characters in them. Preliminary experiments on stemming and lemmatizing the words produced no significant improvements in accuracy.

Translational probabilities do not play any role in our framework. The translation lexicon is therefore regarded as a table which maps one word in the source language to one or more words in the target language. There are two ways to obtain such a translation lexicon with one-to-many entries. One option is to train it automatically from a parallel corpus [17] and ignore (or threshold) translational probabilities. However, it was found that automatically learned translation lexicons contain a considerable amount of noise. There may be dozens of words most of which are actually not associated with the source word. Further, the training process is highly sensitive to the training corpus. A translation lexicon learned from one corpus can not be generalized to another corpus.

A better option is to create a one-to-many translation lexicon using a dictionary. One can make use of all information in the dictionary. All function words are removed on both sides of each entry using a language specific stopword list. If the source entry still consists of multiple words we delete it and do not use it. If the source side of an entry has a single word remaining, then one should include it in the translation lexicon along with all its possible translations one after the other. If a source word maps translates to multiple words then each of these possible translations is listed one after the other in the sequence. If the source word maps to a phrase, the phrase is split into separate words and every word in the phrase is listed as a possible translation in the hope that one of them will map correctly. If more than one dictionary is available, one can also create a larger dictionary by merging translation entries.

3.4 Sequence Alignment

After the translating the unique word sequences of books in the source language to the target language, the next step is to compare each of them against all the books in the target language. Comparison is performed using the Longest Common Subsequence (LCS) algorithm. LCS is basically a global alignment method which gives the longest sequence preserving the long range order between two sequences. Having a large number of words in common preserving the order is a clear indication of translation.

There are a number of algorithms to compute LCS in the literature [8]. The standard dynamic programming algorithm has $O(mn)$ time and space requirements, where m and n are the lengths of the input sequences. For long input sequences, this algorithm has very large memory requirements. Therefore we adopt an $O(mn)$ time and linear space LCS algorithm [13] to calculate the LCS length without computing the actual LCS sequence itself. There is also a $O(nloglogn)$ time LCS algorithm for sequences where no element appears more than once within either input string [15]. This algorithm is not suitable for our purposes because the translated word sequence may include repeated words.

There are further improvements for fast LCS computation. It is not necessary to compute LCS over the entire input sequences. One can disregard the words which do not appear in both sequences since a word must appear in both sequences at least once in order to be in the LCS. Another improvement is to avoid LCS computation entirely when conditions apply. Given the score threshold (used for classifying books pairs to be translations) and the lengths of the sequences, it is possible to solve for a lower bound for the LCS length L. If the number of common words between two sequences is less than L, then there is no need for the alignment procedure since the resulting score is guaranteed to be lower than the threshold. These improvements provide significant speed-up. It should be noted that the intersection of elements between two sequences can be computed in linear time using a hashtable.

3.5 Scoring Functions

The length of LCS between the list of translated words and the list of unique words is used to classify or rank translation pairs. The LCS length alone can not be used for translation detection. The reason is that the number of unique words (hence the length of LCS) is a function of the book length according to Zipf's Law. Longer texts are expected to have longer lists of unique words. It is therefore desirable to normalize the LCS length based on the size of the books compared. Here we adopt the normalization techniques proposed in [30]. These approaches are elaborated in the subsections.

3.5.1 Correlation Score (TRANS-cs)

Using the analogy with correlation, the TRANS-cs score for two sequences of words X and Y is defined similar to the DUPNIQ-cs score in [30] as:

$$TRANS-cs(X,Y)=\frac{|LCS(X,Y)|}{\sqrt{|X||Y|}} \tag{1}$$

where $|LCS(X,Y)|$ is the LCS length for the aligned sequences. $|X|$ and $|Y|$ represents the length of X and Y respectively. The resulting score has a range of [0,1]. The score is maximized when the two sequences are identical.

3.5.2 Information Theoretic Score (TRANS-its)

In this context, input word sequences are defined as objects X and Y and those objects are assumed to be generated by a probabilistic model. Then, according to Lin [20], the similarity between any two objects can be defined as:

$$similarity(X,Y)=\frac{\log\Pr(common(X,Y))}{\log\Pr(description(X,Y))} \tag{2}$$

Similarity is maximized when the two objects are identical. The joint description of two objects is defined to be overall information content of both objects. In our case, the overlapping information content is defined by the longest common subsequence between X and Y and the total information content (description) is defined by the alignment produced by LCS. Once the probability of any word sequence is assumed to be inversely proportional to its length, then Lin's equation simplifies as:

$$TRANS-its(X,Y)=\frac{\log|LCS(X,Y)|}{\log\left(|X|+|Y|-|LCS(X,Y)|\right)} \tag{3}$$

TRANS-its has a range of [0,1]. The score is assumed to be zero if input sequences have no common words.

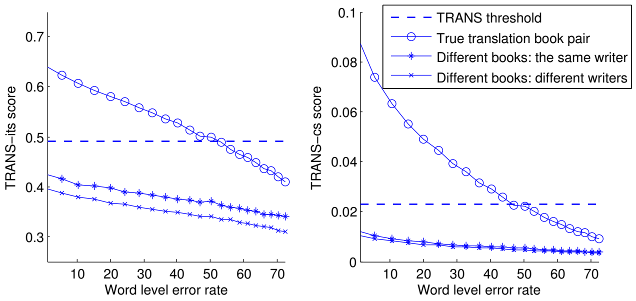

Figure 3: The effect of OCR errors on the translation scores are investigated for three different scenarios. TRANS-its (left) and TRANS-cs (right) scores are shown as a function of word level synthetic document noise. Both measures are able to classify the book pairs correctly for the given thresholds even for high rates of character level document noise.

4. SYNTHETIC EXPERIMENTS

We investigate the effect of OCR errors on translation detection by generating synthetic errors in texts. A pair of texts is created as follows: Two error-free (no OCR errors) books are downloaded from the Project Gutenberg website [2] - one in the source language (the reference text) and a second in the target language. The latter is used for generating synthetic texts by adding a specified amount of random character level document noise to simulate OCR errors. Unique words in the reference text are translated in to the target language. TRANS-its and TRANS-cs scores are computed for the reference and synthetic texts for different levels of document noise from 0% to 20% with 1% increments. Experiments are repeated one hundred times - each time with different random seeds - and the scores are averaged.

The noise model introduced in [10] is adopted for generating the synthetic texts. The model basically performs string edit operations (insertion, deletion and replacement) over the entire text for the given amount for each type of noise. The total amount of noise is defined to be the total percentage of characters deleted, replaced and inserted over the entire string. The distribution of edit operations is defined to be uniform, i.e., [1/3, 1/3, 1/3] respectively. Case is folded and all punctuations and numerals are removed. The English-German dictionary used in the synthetic experiments contains 62K words including inflections.

Three different scenarios are investigated. In the first scenario, we evaluate the effect of OCR errors for true translation pairs. In this case, the reference book is chosen to be “Egmont” which is written in German by Johann Wolfgang von Goethe and synthetic texts are generated using the English translation of the same book. In the second scenario, the same process is applied to two different books which are known not to be translations of each other but written by the same author - the German original of Goethe’s “Egmont” and an English translation of ‘Goethe’s “Faust”. The purpose of this scenario is to test the robustness of the proposed method for texts having similar style and vocabulary. The third scenario investigates the case in which two different books are written by different authors - the German version of Goethe’s Egmont and an English version of “The Critique of Pure Reason” by Immanuel Kant. In a collection the most common scenario is one where the books are not translations of each other and the authors are also different.

In Figure 3, it is clear that TRANS-its and TRANS-cs scores are substantially larger for the true translation pair compared to the other two non-translation pairs. For all scenarios, the translation scores are the highest when there is no document noise and they gradually fall as the amount of noise is increased. TRANS-cs score tend to fall more drastically compared to TRANS-its. For the true translation pair, TRANS-its and TRANS-cs scores fall below the given thresholds at approximate word error rate levels 49% and 44% respectively. Notice that these word error rates are very high and unlikely to happen in practice for printed books. [31] estimate that the OCR word error rate of scanned books in the IA database is less than 15% . The proposed method is robust to the OCR errors found in scanned book collections.

Table 1 provides further detail. In all scenarios, it is seen that the number of unique words increases as the amount of noise increases. The reason is that document noise (or OCR errors) tend to produce arbitrary words which are not in the vocabulary of the book (or even the language).

It is seen that the non-translation book pair having the same author has more common words and higher translation scores compared to the third scenario where the non-translation book pair has different authors. The reason is that different books written by the same author are likely to have more common words in the vocabulary, even though one of them is translated by someone else. Despite this effect, the proposed method successfully discriminates both non-translation book pairs from the true translation pair.

The length of the sequence of words following the same order in both contexts is a clear indication of translation. This can be seen more clearly for the book pairs having the same writer (scenarios 1 and 2). See Table 1. Both book pairs have comparable numbers of common words in their representations. This information alone does not help discriminate these two cases. However, the length of the LCS is considerably higher for the true translation pair. This means that there are a large number of words following the same order for the true translation pair whereas it is not the case for the other. The sequence information of words is therefore a strong feature to detect translations. It is sufficient to have a small number of words in common preserving the same order compared to the total number of unique words in the book.

5. EVALUATION METRICS

Three different evaluation methods are defined to elucidate different aspects of the problem and also depending on what kind of ground truth is available. For large datasets, it is not possible to obtain manually labeled ground truth. In such cases, a retrieval approach must be adopted as described below.

Retrieval of Translations: In this approach, each book in the source language (English in our example) is regarded as a query and all the books written in the target language (German) are ranked according to their translational similarity score. MAP (Mean Average Precision) is calculated over the rank lists. The retrieval approach is feasible especially for large datasets since the evaluation is practical. One can adopt a pooling approach in analogy with the traditional IR ranking paradigm to obtain relevance judgments. The details are described in the experimental section.

Ranking All Book Pairs: Krstovski & Smith [19] rank all the book pairs in a single list according to some simi-

Table 1: Detailed statistics for the three pairs of books examined in Figure 3. $|X|$ and $|Y|$ corresponds to the number of unique words in books X and Y respectively. $|X \cap Y|$ corresponds to the number of common words between |X| and |Y| without any translation. $|X_T \cap Y|$ refers to the number of common words after translating the words in |X| to the language of book $|Y|$. $|LCS|$ is the length of the longest common subsequence between the word sequence representations. TRANS-its and TRANS-cs scores are also shown.

Char err.	Word err.	English	German						TRANS	TRANS
rate(%)	rate (%)	Book X	Book Y	|X|	|Y|	$\|X \cap Y\|$	$\|X_T \cap Y\|$	|LCS|	its	cs
0	0	Egmont	Egmont	2395	3224	32	492	251	0.643	0.090
1	5.33	Egmont	Egmont	2395	4232	34	474	235	0.623	0.074
3	15.29	Egmont	Egmont	2395	5109	34	438	209	0.593	0.055
5	24.38	Egmont	Egmont	2395	7395	33	455	187	0.570	0.044
10	43.68	Egmont	Egmont	2395	10256	30	337	128	0.514	0.026
0	0	Faust	Egmont	3706	3224	27	416	43	0.426	0.012
1	5.33	Faust	Egmont	3706	4232	29	406	42	0.415	0.011
3	15.29	Faust	Egmont	3706	5109	35	388	40	0.401	0.008
5	24.38	Faust	Egmont	3706	7395	37	363	37	0.388	0.007
10	43.68	Faust	Egmont	3706	10256	41	306	35	0.374	0.006
0	0	Kant	Egmont	2625	3224	6	270	31	0.396	0.011
1	5.33	Kant	Egmont	2625	4232	9	263	30	0.387	0.009
3	15.29	Kant	Egmont	2625	5109	10	250	30	0.374	0.007
5	24.38	Kant	Egmont	2625	7395	11	236	29	0.364	0.007
10	43.68	Kant	Egmont	2625	10256	12	205	26	0.345	0.005

larity score and compute Average Precision (AP) over the entire ranked list. This is different than the retrieval of translations approach. Consider the following list of English books E1, E2, E3 and German books G1, G2. Assume that the following ranked list is produced after comparing all the source-target book pairs (E3G1, E1G2, E2G2, E1G1, E3G2, E2G1). The retrieval of translations approach instead use E1, E2 and E3 as queries and compute the AP for each ranked list (E1G2, E1G1), (E2G2, E2G1) and (E3G1, E3G2) and average all the AP values to compute a MAP score. The ranking all book pairs approach is reasonable as long as the ground truth for the entire dataset is available. One may still go over the entire ranked list and annotate each pair manually. However, this is not feasible for large datasets since the number of book pairs to be checked is significantly larger than for the retrieval approach.

Binary Classification: This measure requires the system to classify each book pair as a translation or not. In the approaches we use this is done using a threshold over the translation scores. If the ground truth is available for the entire dataset, then precision and recall values can be generated. It should be noted that precision/recall values are the most restrictive metrics, since they require translational scores to be comparable between different book pairs and a careful selection of the score threshold. Even if MAP and AP scores are both 1.0, it is possible to get either precision or recall values below 1.0. It happens when the score threshold is either too high or too low. The least restrictive evaluation metric is the MAP score for the retrieval task since it does not require the translational scores to be comparable between different queries.

6. EXPERIMENTS

This section begins with a listing of the datasets collected and used. This is followed by a description of the translation lexicons used. Following this is a discussion of the baselines and other algorithms used for comparison. Finally, we describe a set of experiments carried out and the results obtained from them.

6.1 Datasets

Books downloaded from the Internet Archive (IA) [1] were used to construct datasets. English-German training and the 2K datasets are publicly available [2].

An English-German training set contains 30 scanned books (16 English, 14 German) from the IA database. It is manually verified that a book has at least one translation in the set. There are 31 true translation pairs in total. This set is used to estimate the translational similarity threshold for the scanned book experiments.

The EUROPARL parallel corpus is a standard collection of text documents from the proceedings of the European Parliament [16] used for machine translation. These documents are clean - since they have no OCR errors. Version 3 is used for our experiments in order to compare the results with the baseline approach described in [19]. It contains speeches from the period 04/1996-10/2006. There are over 600 documents each of which is translated in to 11 languages. Unlike the scanned book collections, these texts do not include any document noise since they are translated and typed by humans. Among these parallel corpora, we use four language-pairs: English-Finnish, English-French, English-German and English-Spanish. Notice that Finnish is from a different language family compared to the other languages. The average number of words per document in the English collection is 50360 after removing the tags. Many of these documents are much shorter than most books.

The 2K dataset is an English-German collection of 2K scanned books and is one of the datasets used by Krstovski & Smith in [19] and they refer it as the “17 book pairs” dataset. The dataset is originally created by downloading a random selection of 1K German and 1K English books from the IA website and embedding 17 book translation pairs in it. However, our approach discovered that there are actually 18 translation book pairs in the dataset. TRANS found three additional translation pairs and falsified two translation pairs which were initially in the ground truth. After

[2] http://ciir.cs.umass.edu/downloads/trans-detect/ and http://books.cs.umass.edu/downloads/trans-detect/

Table 2: Dictionary statistics after ignoring phrasal translations.

Dictionary	Words	Translation Success
English-German 62K	62242	79.8%
English-German 5K	5487	19.7%
English-Finnish	2997	11.9%
English-French	17326	54.2%
English-Spanish	23377	53.2%

manual investigation, the ground truth for this dataset has been corrected and it is used for the experiments along with the updated results obtained from Krstovski & Smith.

The 50K dataset is a collection of 50K books in German randomly selected from the the IA database. Using the language identifier, it is verified that the OCR outputs are not garbage and that the dominant language of these texts is German. This set is used only for ranking experiments. A set of 20 famous books in English are used for querying. Query books are chosen in a way that there exists at least one translation for each of them in the entire collection. The ground truth for the query set is obtained as follows: for each query book, books in the 50K collection are ranked according to the TRANS-cs, TRANS-its and metadata scores. Each of these techniques produces a ranked list for each query. The top 200 ranking entries from all three lists were pooled for each query and then manually judged. This pooling approach provide a basis to determine relative effectiveness of the systems being compared. In total, 52 translation pairs were labeled for all 20 queries.

6.2 Translation Lexicons

There are two ways to obtain a translation lexicon. The first one is to learn translations from a parallel corpus. The second one is to use a dictionary. We first tried to learn a translation lexicon for the English-German language pair using a statistical machine translation system [17]. Training was performed on the Europarl parallel corpus. However, final precision and recall figures were quite low compared to the dictionary approach. Therefore we decided to use the dictionary approach for the rest of our experiments.

Table 2 below shows statistics on the size of the dictionaries used in our experiments [3]. All the dictionaries provide translations for different forms of the word (such as plural, gerund, past participle etc.), whereas the English-German 5K and English-Finnish dictionaries lack this feature. We also provide the average percentage of unique words translated using each dictionary. The percentages are generated for the EUROPARL corpus. We also tried a number of lemmatization techniques in order to improve translation success. Even if we observed improvements in the total number of translated words, no improvement is observed in the precision and recall figures. Dictionary size and OCR error rate are the determinants of the overall success of the framework.

6.3 Baselines

Most work on creating parallel corpora has been focused on small datasets and using either structural information or the alignment of individual sentences [28] with two exceptions: Uszkoreit et al. [29] and Krstovski & Smith [19]. Uszkoreit's approach is not used as a baseline since the datasets and the translation system they used are not available to us. Here we use three baseline systems: metadata search, IBM MODEL 1 and where available numbers from Krstovski & Smith [19].

META refers to using metadata search for finding translation pairs in a collection of books. Here we use title and author information from the IA database as follows: first all the punctuation in the author and title fields are removed and all the characters are lowercased. Numeric characters are also ignored only for the author field since the date information leads to false matches. The title of the query book is also translated from English to German using the Google Translate API. The set of tokens in the author field of the query book is compared against the books in the collection of 50K German books using the Jaccard similarity. If the similarity is greater than zero, then the translated title is also compared against the title of each candidate book in the same way. The "metadata score" for a single pair of books is defined to be the average of the title and author Jaccard similarities. The metadata score is used to detect/rank books pairs for being translations. Notice that the metadata is not fully reliable since it is typed by people who scan and/or upload the book in to the IA database.

IBM M1 refers to the widely-used IBM Model 1 used for aligning words given two sentences in different languages [6]. It is used for different tasks over parallel corpora and essentially gives an estimate for the probability of a target sentence T in some language given a source sentence S in another language. There are several simplifying assumptions in this model. It does not incorporate any information about the long range order of words in the source and target sentences unlike the sequence of unique words. This approach is therefore ideal to demonstrate the effectiveness of bag-of-words models over long texts. Since this model is effective for ranking, we use it only for retrieval and ranking experiments. For fairness, the same dictionary is used for all techniques. Transition probabilities are estimated by assuming that all translations are equiprobable.

Krstovski & Smith use an approach for generating a ranked list of book translation pairs without the use of bilingual dictionary or machine translation system [19]. Each book in the collection is represented in the vector space and cosine similarity is used to rank all the book pairs in the collection. The vector representation only accounts for the words which appear in both languages without any translation. For each book, the weights of the vector representation are calculated by multiplying the frequency of the term in the book with the inverse document frequency of the term in the collection of books in the same language, i.e. (TFx-IDF). The Locality sensitive hashing (LSH) approximation algorithm is used to calculate cosine similarity to reduce the time complexity. We use their datasets and results which are publicly available.

6.4 EUROPARL Experiments

The EUROPARL dataset is used to test the effectiveness of our approach for documents with no OCR errors. There are roughly 650 documents per language each of which has a translation in the other language. For each language pair we selected 50 translation pairs at random as a training set and used the remaining as a test set. The training set is used to train the score threshold (a different threshold for each language since dictionary sizes vary significantly). For English-German, the 62K dictionary is used. The evaluations are

Table 3: Translation retrieval experiments for the EUROPARL dataset.

Dataset	TRANS-its	TRANS-cs
	MAP	MAP
Eng-Fin	1.0	1.0
Eng-Fre	1.0	1.0
Eng-Ger	1.0	1.0
Eng-Spa	1.0	1.0

Table 4: Ranking all document pairs for the EUROPARL dataset.

Dataset	TRANS-its AP	TRANS-cs AP	Krs.&Smith AP
Eng-Fin	1.0	1.0	-
Eng-Fre	1.0	1.0	-
Eng-Ger	1.0	0.994	0.986
Eng-Spa	1.0	1.0	-

done on the test set. The retrieval and ranking all pairs experiments are shown in Tables 3 and 4 respectively. Binary classification results are given in Table 5. We notice that TRANS-its has a MAP score of 1.0 and an AP of 1.0 for both the retrieval and ranking of all pairs evaluations. TRANS-cs performs slightly worse on the English-German ranking of all pairs evaluation. We also list Krstovski&Smith's result for the English-German pair from their paper (their splits are different but the results are indicative). Krstovski&Smith do not provide numbers for the other language pairs but they have graphs which clearly show that the AP score must be less than 1.0.

The binary classification experiments indicate that threshold selection is a hard problem compared to the ranking and retrieval paradigms. TRANS-its has a precision of 1.0 for all language pairs. TRANS-its also ranks all the document pairs perfectly since the AP score is 1.0. However, the recall values are slightly lower than 1.0 for English-Finnish and English-German datasets. The reason is that one pair in the English-Finnish and two pairs in the English-German dataset are below the score threshold although they are relevant. Further analysis of the results show that missing document pairs are actually very short (a few hundred words). Our technique is quite robust for longer documents. Precision and recall values for TRANS-cs are both lower than 1.0 for the English-German dataset, which indicates there are relevant documents below the score threshold while there are false positives with a score higher than the threshold. Clearly TRANS-its performs very well on all metrics.

6.5 Experiments with Real Scanned Books

Table 6 shows results for the retrieval experiments on real scanned books for the English-German datasets (Train, 2K and 50K). The best scores are shown in bold face. TRANS-its (using the large dictionary) is the most successful system among all others providing MAP scores of 1.0. Note that the results are worse when the smaller dictionary is used. Metadata search (META) performs well in ranking books for both the train and test sets but not as well for the 50K dataset (MAP = 0.821). TRANS-cs is much worse indicating the importance of LCS length normalization. IBM-M1 performs poorly. In all the tables below "Dict" refers to the size of the dictionary.

Table 5: Classification experiments for the EUROPARL dataset.

Dataset	TRANS-its		TRANS-cs	
	Precision	Recall	Precision	Recall
Eng-Fin	1.0	0.998	0.973	1.0
Eng-Fre	1.0	1.0	1.0	1.0
Eng-Ger	1.0	0.997	0.992	0.995
Eng-Spa	1.0	1.0	0.992	1.0

The evaluation for the ranking all book pairs experiment is given in Table 7. Experiments are not performed on the 50K dataset because it would require judging several thousand entries. TRANS-its again has an AP of 1.0 for the train and 2K datasets. Metadata search has a lower AP scores on both train and 2K datasets. IBM-M1 again performs poorly. Note that TRANS-its has an AP score of 1.0 even with a dictionary of size 5K. TRANS-cs performs slightly worse with a smaller dictionary. Krs.&Smith obtained an AP = 0.945 for the 2K dataset (their precision, recall and MAP results are not available for the 2K dataset).

Binary classification is performed by learning the score threshold from the train set and it is used for the 2K dataset. As seen in Table 8, TRANS-its with 62K dictionary gives perfect precision and recall values for both datasets. TRANS-cs and TRANS-its both provide perfect scores on the train set even with a small dictionary. Precision values for the 2K dataset fall if the small dictionary is used. The drastic fall in the precision figures for the 2K dataset is due to the low score threshold. This indicates that there is a need for a better threshold selection paradigm since both score functions actually perform very well in ranking all book pairs experiment, as shown in Table 7. Surprisingly metadata search does not provide perfect scores (precision = 0.739, recall = 0.944) for either the 2K set or the train set.

Uszkoreit et al. [29] best published result (using an oracle to choose the threshold) for a dataset of 103 books (English-French) with 30 matching pairs has a precision of 1.0 and a recall of 0.71. Although it is not directly comparable, we note that TRANS-its has both precision and recall 1.0 on a 2K book dataset. TRANS also does not require complete translation of books. Unfortunately, their machine translation system and datasets are not publicly available for us to be able to make a direct comparison.

7. CONCLUSIONS

A translation identification framework is presented for large scanned book collections with OCR errors. Unique words (which appear only once in the whole book) along with their actual order in the text are used to represent each book in the collection. This sampling strategy provides a compact representation and it enables efficient identification of translation pairs. A dictionary approach is adopted to translate word sequence representations. Fairly small dictionaries work well. The proposed approach is shown to be quite robust to high rates of OCR errors and it outperforms several baselines including metadata search. Retrieval experiments on several datasets including the Europarl parallel corpus with four different language pairs show that the proposed method retrieves translation pairs with a MAP score of 1.0. Future work includes further speed-ups, extensions to multiple languages and mapping translated portions.

Table 6: MAP scores for the retrieval experiments on scanned book datasets.

Approach	Dict	Dataset		
		Train	2K	50K
TRANS-its	62K	**1.0**	**1.0**	**1.0**
TRANS-cs	62K	1.0	1.0	0.717
TRANS-its	5K	1.0	1.0	0.714
TRANS-cs	5K	1.0	1.0	0.669
META	-	0.99	1.0	0.821
IBM-M1	62K	0.302	0.008	< 0.001

Table 7: AP scores for ranking all book pairs experiments for scanned book datasets.

Approach	Dict	Dataset	
		Train	2K
TRANS-its	62K	**1.0**	**1.0**
TRANS-cs	62K	1.0	1.0
TRANS-its	5K	1.0	1.0
TRANS-cs	5K	1.0	0.943
META	-	0.959	0.916
IBM-M1	62K	0.148	0.0002
Krs.&Smith	-	-	0.945

8. ACKNOWLEDGMENTS

We would like to thank David Smith and Kriste Krstovski for providing their datasets and results, Ethem F. Can for helping with the figures and Internet Archive for their book collection. This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

9. REFERENCES

[1] Internet Archive. http://www.archive.org, 2012.

[2] Project Gutenberg. http://www.gutenberg.org, 2012.

[3] Wordgumbo:. http://www.wordgumbo.com, 2012.

[4] Y. Bernstein and J. Zobel. A scalable system for identifying co-derivative documents. In *SPIRE*, pages 55–67, 2004.

[5] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. *Computer Networks*, 29(8-13):1157–1166, 1997.

[6] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer. The mathematics of statistical machine translation: parameter estimation. *Comp. Ling.*, 19:263–311, June 1993.

[7] P. Clough. Old and new challenges in automatic plagiarism detection. National UK Plagiarism Advisory Service, http://www.ir.shef.ac.uk/cloughie/papers/pas_plagiarism.pdf, 2003.

[8] S. Deorowicz. Solving longest common subsequence and related problems on graphical processing units. *Softw. Pract. Exper.*, 40:673–700, July 2010.

[9] M. Errami, Z. Sun, A. C. George, T. C. Long, M. A. Skinner, J. D. Wren, and H. R. Garner. Identifying duplicate content using statistically improbable phrases. *Bioinformatics*, 26(11):1453–1457, 2010.

[10] S. Feng and R. Manmatha. A hierarchical, HMM-based automatic evaluation of OCR accuracy for a digital library of books. In *JCDL*, pages 109–118, 2006.

[11] P. Fung. Compiling bilingual lexicon entries from a non-parallel english-chinese corpus. In *Annual Meeting of Very Large Corpora*, 1995.

Table 8: Binary classification results on English-German datasets. "Thr", "P", "R" are threshold, precision and recall respectively.

Approach	Dict	Thr	Train		2K	
			P	R	P	R
TRANS-its	62K	0.49	**1.0**	**1.0**	**1.0**	**1.0**
TRANS-cs	62K	0.023	1.0	1.0	0.782	1.0
TRANS-its	5K	0.395	1.0	1.0	0.122	1.0
TRANS-cs	5K	0.0085	1.0	1.0	0.01	1.0
META	-	0.275	0.882	0.968	0.739	0.944

[12] G. Grefenstette. Comparing two language identification schemes. In *3rd International Conference on Statistical Analysis of Textual Data (JADT 95)*, December 1995.

[13] D. S. Hirschberg. Algorithms for the longest common subsequence problem. *J. ACM*, 24:664–675, October 1977.

[14] T. C. Hoad and J. Zobel. Methods for identifying versioned and plagiarized documents. *JASIST*, 54(3):203–215, 2003.

[15] J. W. Hunt and T. G. Szymanski. A fast algorithm for computing longest common subsequences. *Commun. ACM*, 20:350–353, May 1977.

[16] P. Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *The tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT.

[17] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *ACL*, 2007.

[18] K. Koroutchev and M. Cebri. Detecting the same text in different languages. In *IEEE Information Theory Workshop ITW*, pages 337–341, 2007.

[19] K. Krstovski and D. A. Smith. A minimally supervised approach for detecting and ranking document translation pairs. In *6th Workshop on SMT*, pages 207–216, 2011.

[20] D. Lin. An information-theoretic definition of similarity. In *ICML '98*, pages 296–304, 1998.

[21] X. Ma and M. Liberman. Bits: A method for bilingual text search over the web. In *Machine Trans. Summit VII*, 1999.

[22] D. Mimno, G. Crane, and A. Jones. Hierarchical catalog records: Implementing a FRBR catalog. *D-Lib Magazine*, 11(10), Oct 2005.

[23] J. Y. Nie, M. Simard, P. Isabelle, and R. Durand. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In *ACM SIGIR*, pages 74–81, 1999.

[24] M. Potthast, A. Barrón-Cedeño, B. Stein, and P. Rosso. Cross-language plagiarism detection. *Language Resources and Evaluation*, 45(1):45–62, 2011.

[25] P. Resnick and N. Smith. The web as a parallel corpus. *Computational Linguistics*, 29(3), 2003.

[26] P. Resnik. Mining the web for bilingual text. In *ACL*, pages 527–534, 1999.

[27] N. Shivakumar and H. Garcia-Molina. Scam: A copy detection mechanism for digital documents. In *Ann. Conf. on the Theory and Practice of Digital Libraries*, 1995.

[28] N. Smith. From words to corpora: Recognizing translation. In *EMNLP*, pages 95–102, 2002.

[29] J. Uszkoreit, J. Ponte, A. C. Popat, and M. Dubiner. Large scale parallel document mining for machine translation. In *COLING*, pages 1101–1109, 2010.

[30] I. Z. Yalniz, E. F. Can, and R. Manmatha. Partial duplicate detection for large book collections. In *CIKM*, pages 469–474, 2011.

[31] I. Z. Yalniz and R. Manmatha. A fast alignment scheme for automatic ocr evaluation of books. In *ICDAR*, pages 754–758, 2011.

[32] C. C. Yang and K. W. Li. Automatic construction of english/chinese parallel corpora. *JASIS*, pages 730–742, 2003.