INFORMATICA, 2018, Vol. 29, No. 4, 693–710

DOI: http://dx.doi.org/10.15388/Informatica.2018.188

Sentence Level Alignment of Digitized Books Parallel Corpora

Algirdas LAUKAITIS[1]*, Darius PLIKYNAS[2], Egidijus OSTASIUS[1]

[1]*Vilnius Gediminas Technical University, Fundamental Science Faculty*
[2]*Vilnius University, Institute of Data Science and Digital Technologies*
e-mail: algirdas.laukaitis@vgtu.lt, darius.plikynas@mii.vu.lt, egidijus.ostasius@vgtu.lt

Received: September 2017; accepted: September 2018

Abstract. In this paper, we propose a framework for extracting translation memory from a corpus of fiction and non-fiction books. In recent years, there have been several proposals to align bilingual corpus and extract translation memory from legal and technical documents. Yet, when it comes to an alignment of the corpus of translated fiction and non-fiction books, the existing alignment algorithms give low precision results. In order to solve this low precision problem, we propose a new method that incorporates existing alignment algorithms with proactive learning approach. We define several feature functions that are used to build two classifiers for text filtering and alignment. We report results on English-Lithuanian language pair and on bilingual corpus from 200 books. We demonstrate a significant improvement in alignment accuracy over currently available alignment systems.

Key words: alignment of corpora, alignment of digitized books, machine translation, natural language processing.

1. Introduction

Translation memory extraction is the problem of extracting word and phrase translations from bilingual corpora. These translations are usually extracted from technical document parallel corpora, which can be easily aligned on sentence level with low error rates. However, for most language pairs there isn't a sufficient amount of technical document parallel data in order to build high quality translation memory database.

In order to extend translation memory database there have been numerous approaches to extract parallel sentences from non-parallel monolingual corpus, such as news articles or web pages. While these methods have been applied for translation dictionary improvement, little attention has been given to a corpus of translated books as a source of bilingual sentences. This is surprising given that the corpus of translated books has potential for extending translation memory much better than non-parallel monolingual corpus.

As an example, we can look at the corpus of English-Lithuanian language pair that we created in the past few years. The corpus consists of two parts: 1.8 million sentences

*Corresponding author.

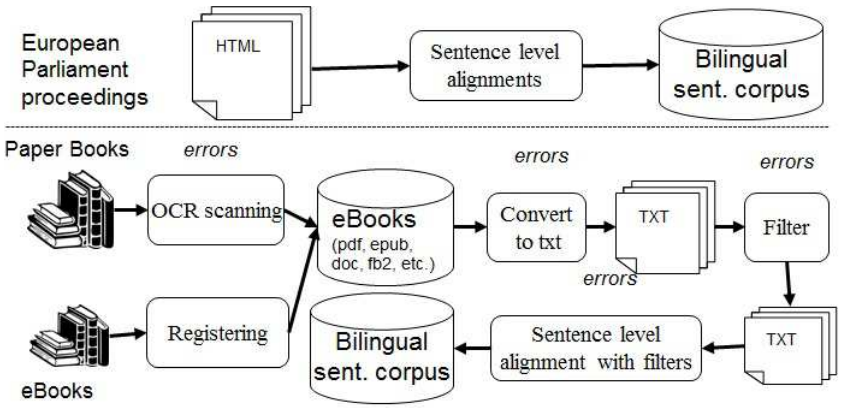

Fig. 1. Differences between technical documents' and books' alignment processes.

from the European Parliament proceedings and 2.1 million sentences from the corpus of 200 books. The domain range of the books' corpus is more complex and its language is more expressive than the language from European Parliament proceedings. Yet, these 200 books represent only a small fraction of all the books that have translations in English and Lithuanian languages.

We can ask ourselves why there was so little research done in order to create methods and algorithms for bilingual book corpus alignment. In order to answer this question we can look at Fig. 1, where we compare differences in alignment processes of technical bilingual corpus vs corpus consisting of fiction and non-fiction books.

In traditional technical document alignment process, we already have well formatted document pair and run alignment procedure which is based on one of the sentence length based alignment algorithms. Usually we get high quality alignments by using sentence length metric alone without any additional preprocessing. By adding some additional lexical information, i.e. dictionaries, minor improvements in alignment quality can be achieved.

On the other hand, we can see from Fig. 1 that the books alignment process is more complex. There are at least four factors of additional string generation process (these strings we need to detect and remove) when we try to align books corpus. First, in some cases we must use optical character recognition (OCR) process in order to get text from a book. This process usually generates errors determined by OCR software. Even if we use the state-of-the-art OCR systems we still can get some errors. So, we usually prefer an alternative source to ORC systems, i.e. books in some digital format, i.e. *pdf, epub, fb2, doc*, etc.

The next process that can produce additional erroneous strings for alignment algorithms is due to the requirement to transform various document formats (e.g. *pdf, epub, fb2, doc*) to a single text format which can be used for the final alignment. Many file format converters are available, but it can happen that for some books they produce a transformation error. We found that the type of these errors is similar to error type we get from OCR process. Thus, we can tackle these two transformation error types with the same classifier.

After we get a text file, we need to use filters in order to remove various page formatting marks and notes that were entered by the editor and the translator (e.g. translator notes at the end of book page can be a significant source of errors). We would like to remove these insertions and for this we developed filters based on regular expressions.

Once we get the filtered text files we still need a special algorithm for book alignment because existing algorithms usually work well only on relatively small documents. We found that the major factor of alignment errors at this stage of corpus processing is due to a very figurative translation of some books. Sometimes the translator can skip several paragraphs just because some passages in book require substantial efforts to translate. What we would like to do in this case is to identify such omissions and mark them to be excluded from alignment process.

These introductory remarks define the rest of the paper, which is organized as follows. In the next section, we review some related works and discuss a few open issues. In Section 3 we present the general architecture of the system for the bilingual books' corpus creation. Section 4 describes statistical inference model for finding chapter, paragraph and sentence punctuation marks based on conditional random field model. Section 5 describes filtering algorithms for correcting alignments. In Section 6 we describe the various elements of user interaction model. Our view is that there is always the likelihood that the alignment algorithm can be stuck in a local minimum. In this case we suggest proactive learning model to query for an alignment anchor point. Section 7 describes empirical tests of the suggested method. Finally, concluding remarks and future work are presented at the end of the paper.

2. Related Work

Several techniques have been proposed to align corpus at sentence level. Techniques that are unsupervised and language independent use mainly sentence length statistics in order to find relationships between sentences that are translations of each other. One of the first methods that used this approach was reported in Brown *et al.* (1991). It achieved an accuracy of 99% in a random selected set of 1000 sentence pairs from the English-French Hansard corpora. The method used only the number of tokens in the sentence without any use of the lexical details.

A similar approach but based on a simple statistical model of lengths in characters was proposed by Gale and Church (1993). Their suggested algorithm is based on a probabilistic score of length difference of the two sentences. In both papers dynamic programming techniques are applied to find the maximum likelihood alignment of bilingual sentences.

While these length based algorithms can achieve small error rates on literal translations they are not robust with respect to translations where some sentences are skipped or merged. In order to overcome these challenges, many improvements have been suggested. Chen (1993) developed sentence level alignment algorithm that requires an externally supplied bilingual lexicon. This algorithm gave better accuracy than the length based methods. Moore (2002) proposed a multi-pass search procedure that complements sentence length

based statistics with the IBM Model-1 translation model (Brown *et al.*, 1993). Varga *et al.* (2007) also suggested a similar method. The main improvement over (Moore, 2002) approach is the usage of a word by word dictionary.

A more recent approach reported in Braune and Fraser (2010) used several alignment stages. In the first stage alignments were computed similarly to the method reported in Moore (2002). The second stage uses one-to-one alignments that are obtained through dynamic programming and then the merge procedure was used to obtain many-to-one alignments. The use of an automatic translation system to translate the source sentence for sentence alignment was proposed in Sennrich and Volk (2010). In this work a map of one-to-one alignments is generated based on the BLEU metric. Then various heuristics are used to refine one-to-one and to add many-to-one and one-to-many alignments.

Nevertheless, we found that precision and recall of existing algorithms are too low in order to consider them practical when it comes to an alignment of translated fiction books. As shown in Laukaitis and Vasilecas (2008), Laukaitis *et al.* (2011), the accuracy of these methods decreases drastically when we try to align a text that contains discrepancies, e.g. some book page layout segmentation strings, missing sentences and frequent one-to-many alignments.

Similar conclusions have also been drawn by reevaluation of state-of-the-art methods for large collections of publicly available novels in Xu *et al.* (2015). In this work authors used 24 English-French bilingual books and 17 English and Spanish bilingual books in order to evaluate their method which is based on maximum entropy approach (Berger *et al.*, 1996). They found that by using existing algorithms and the two stage approach one can get slightly better precision than by using methods based on sentence length and lexical features. Our research in this respect on the corpus of 200 books shows that the precision reported in Xu *et al.* (2015) is possible only if books in corpus do not have additional strings, such as translator notes, headers, footers, etc.

We argue in this paper that in order to get high quality alignments on corpus of books we must consider a small number of interactions between the reader of the book and machine learning algorithms. Recent research on computer-assisted translation suggested several methods on how to improve translation using an iterative process in which the human translator interacts with statistical machine translation system (Barrachina *et al.*, 2009). We found that research works in the area of active learning and crowdsourcing can help efficiently build up queries for the book reader. There have been many works in NLP area that investigated active learning approach, such as text classification (McCallum and Nigam, 1998; Tong and Koller, 2001) or information extraction (Thompson *et al.*, 1999; Settles and Craven, 2008). In this paper we suggest how to adopt these active learning strategies for book alignment on sentence level.

3. General Framework

We start our presentation of the system by discussing general framework of alignment process. Many systems that we discussed in the related works section work in two stages.

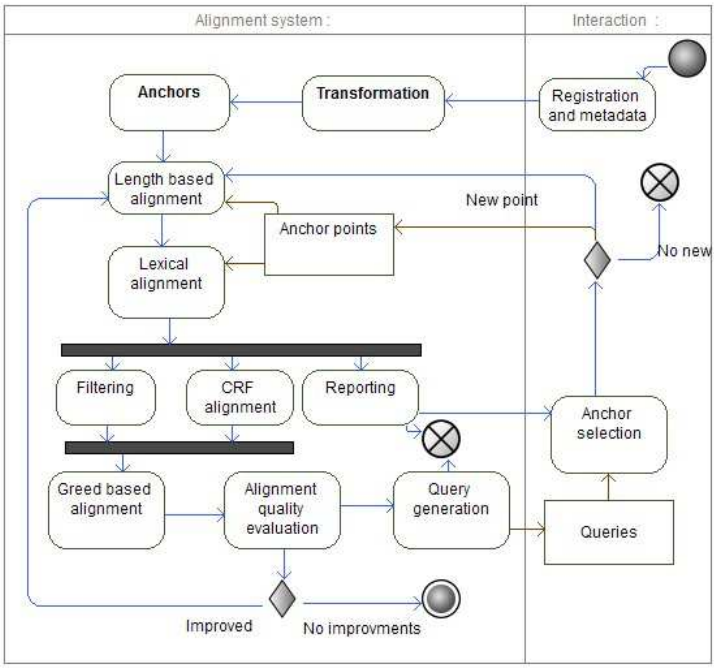

Fig. 2. The general framework for book corpus alignments.

Usually, they perform sparse alignment using lexical features to find rare but highly probable anchor points. Then, these systems try to align corpus between anchor points using length and lexical based approach. As we mentioned above, such approach does not work on book corpus and more processing stages are required.

Therefore, here, we present books alignment general framework through the description of processes that we developed in order to accomplish high quality alignment. In Fig. 2 we can see the UML activity diagram which presents workflow in our alignment framework. We can see that workflow is divided into two parts (swimlanes in the UML diagram): 1) alignment system and 2) interaction system. All activities that require human interaction are put in '*interaction*' swimlane. On the other hand, activities that are processed without reader intervention are put under '*alignment system*' swimlane.

We start with the activity '*registration and metadata*'. The purpose of it is to put a pair of books into the processing pipeline. In our case it means that the reader puts English language book in one file system directory and Lithuanian language book into another directory. These books can be in any popular e-book format: *pdf*, *mobi*, *epub*, etc. Additionally, we encourage the reader to enter some metadata about the book. We try to extract that metadata from wikipedia pages but even if we successfully extract it we still require that the reader confirm that the extracted metadata is correct. This is done because we found that character, location or organization names can be very useful in finding high quality anchor points.

Once we have put books into processing pipeline, we can proceed to carry out the transformation of an e-book from a format like *pdf*, *mobi*, *epub* into the text file format.

There are plenty of free programs and web services that can perform this kind of activity. As an example we can mention that for *pdf* files we use the Apache PDFBox library. We found that in most cases it correctly extracts text from *pdf* files.

After the transformation process we start the alignment process by entering two mandatory anchor points. We require that the beginning and the end of the book have alignment anchor points confirmed by the reader. This means that usually we must trim all the text before the first chapter in the book and all the text after the last sentence of the last chapter. We must do this activity because some e-books (e.g. books form project Gutenberg) have a significant amount of supplementary information that is not related to the context of a book and that is put either at the beginning or at the end of the book.

Once we have these two anchor points, an automatic alignment extraction can be started. Instead of using two stages approach, as many current systems do, we loop several stages of alignment until final convergence by alignment metric is achieved or until reader decides that he is not willing to improve alignments further. We start this loop with two alignment sub-processes: a process that is based on sentence length and then a process that is based on lexical feature functions. The length based sub-process is based on (Gale and Church, 1993) approach. It produces initial set of anchor points. We do not try to align all the sentences but instead we align the group of sentences within a predefined window. Then we try to align segments of a text within each group of sentences using lexical feature functions. In all works that we investigated systems try to use maximum entropy approach (Berger *et al.*, 1996) in order to process this stage. Nevertheless, we found that in many methods (like in Moore, 2002 and Varga *et al.*, 2007) we get more than 50% error rate when we finish these two sub-processes.

Then, all other sub-processes we added in order to achieve error rates that, at the end of alignment, are less than 5%. All these sub-processes start from three threads: conditional random fields (CRF) alignment, filtering and reporting.

Our CRF alignment algorithm is a supplementary component to the traditional alignment method. We developed it in an effort to improve alignment quality and solve a problem that is related to the correlation between feature functions. It is well known that maximum entropy approach creates a label bias problem (Lafferty *et al.*, 2001). In order to avoid this label bias problem we label all alignments from maximum entropy algorithm with the small set of labels. We learn this set of labels from a few manually aligned books using skip-chain conditional random fields (SCCRF). In the next section we present this algorithm in details.

The sentence length and lexical associations baseline alignment sub-processes do not remove footnotes, headers, page numbers, etc. All these additional strings can lead to a complete misalignment in the corpus. For example, let us compare English and Lithuanian translations of the novel 'War and Peace' by the Russian author Leo Tolstoy. In the original Russian text there are many text blocks that are written in French. English language translator decided to translate these French passages into English. Lithuanian language translator decided to leave them in French and add translations in the footnotes. As the consequence of this fact, we got more than 80% alignment error rate when we tried to align this novel translation using existing alignment algorithms.

Therefore, we need to filter out all the elements that have no counterpart in a translated text. During this research project, we developed a text filtering method that helps us to filter out more than 99% strings that have no translations. We found that after this filtering process even the existing methods gained a significant improvement in alignment quality. The output of this filtering process is a set of text segments that our alignment system suggests to delete from books. Section 5 describes this process in more details.

One of the keystones of the alignment model is the reader interaction component, which requires that a set of anchor points be generated after each alignment loop. Very often, due to figurative translation, standard alignment algorithm can be stuck and unable to find good alignments. In this case we need natural language understanding techniques that are beyond capabilities of currently available algorithms. In these cases only a human can help to find a new anchor point from which the alignment algorithm can start to align using sentence length information and lexical entries.

Therefore, the system must ask for a human help. Thus, to complement the automatic alignment loop (left arrow in Fig. 2) we create in parallel another loop for reader interaction. This parallel loop means that reader can intervene into the alignment process at any time he wishes. This loop starts with reporting activity during which alignment system produces comprehensive and highly readable report about current alignment progress. The reader can decide to investigate this report and help automatic alignment system by entering manually some alignment anchors.

The question remains which anchor points are most useful for automatic alignment procedure. We found that it is impractical to show all the context of the book and current alignment state for the reader and ask to find the best alignment in order to help the system. A more intelligent report is required. For this we developed a proactive learning based approach which we present in Section 6.

This approach consists of alignment quality evaluation and query generation after each automatic alignment loop. These queries are sorted by their relevance (information entropy based metric which we discuss in details in Section 6). It is up to the reader to decide which of them to answer. Usually we expect that he will answer at least one query so that the system can evaluate this new information and realign books accordingly.

4. Conditional Random Fields for Sentence Alignment

A linear-chain conditional random fields are undirected graphical models, trained to maximize the conditional probability of the sequence labels (Lafferty *et al.*, 2001). It is well known that the maximum entropy approach could suffer from label bias problems (Lafferty *et al.*, 2001). The CRF models can be used to solve this label bias problem. In this section we present an algorithm that uses linear-chain conditional random fields' models to improve the alignment on sentence level of a bilingual corpus (we use the R package CRF for model training and decoding).

It has been mentioned in previous section that, in our case, the CRF alignment algorithm is a supplementary component to the traditional alignment method. We do not apply

Rays of gentle light shone from her large, timid eyes.	Iš didelių jos akių tryško geros ir baikščios šviesos spinduliai.	liesą veidą ir darė jį gražų. Brolis norėjo paimti paveiksliuką,
Those eyes lit up the whole of her thin,	Šios akys nutvieskė visą ligūstą,	bet ji sulaikė jį. Andrejus suprato,
sickly face and made it beautiful.	liesą veidą ir darė jį gražų.	persižegnojo ir pabučiavo paveikslėlį.
Her brother would have taken the icon,	Brolis norėjo paimti paveiksliuką,	Jo veidas kartu buvo ir švelnus
but she stopped him.	bet ji sulaikė jį.	jis susigraudino)
Andrew understood,	Andrejus suprato,	ir pašaipus.
crossed himself and kissed the icon.	persižegnojo ir pabučiavo paveikslėlį.	Merci, mon ami.
There was a look of tenderness,	Jo veidas kartu buvo ir švelnus	Ji pabučiavo jam į kaktą ir vėl atsisėdo ant sofos.

Fig. 3. Example of positive '*good*' and negative '*bad*' alignments for CRF model training. We randomly shift the target segments in order to get misaligned (negative) examples (third column in the table).

it to get the alignment positions. Instead, we use it to label alignments found by algorithms described in Moore (2002) and Varga *et al.* (2007) with the labels: '*good*', '*bad*'.

As an example, we can look at Fig. 3. In the first and second columns we can see a few English and Lithuanian sentences from the novel 'War and Peace' by the Russian author Leo Tolstoy. These pairs of sentences represent alignments that have been verified by the reader and labelled as 'good'. On the other hand, the third column represents distorted alignments that have been misaligned intentionally by randomly shifting and joining some 'good' alignments. We label them as 'bad' alignments.

We start our analysis from the set of variables e, l, a. Variable e represents the English language sentences and variable l represents translation sentences in the Lithuanian language. A hidden alignment variables a describes a mapping between sentence punctuation marks. The relationship between these variables in the statistical machine translation model is defined by

$$P(e|l) = \sum_{a} P(e, a|l). \tag{1}$$

In order to find variables a we must consider the optimization problem

$$\hat{a} = \arg\max_{a} P(e, a|l), \tag{2}$$

where $\hat{a}$ is alignment that has the highest probability and is called Viterbi alignment. This generative model for finding alignments can be used when we try to find alignments on word level and when we have at least several million of sentences parallel corpora. But, as we already mentioned, for the books corpus this model doesn't give a required recall and precision. In order to align books on sentence punctuation marks we can use maximum entropy models. These discriminative models use conditional probabilities and relevant feature functions. The drawback of maximum entropy model (Berger *et al.*, 1996) is that it makes decisions at each alignment point independently.

Nevertheless, we found that maximum entropy model alignment precision can be greatly improved if we try to align relatively small fragments between fixed anchor points from books' corpus. That was why we decided to try CRF models to label alignments and then to use 'good' alignments as anchor points in the next alignment iteration.

Linear-chain conditional random fields' model lets us use correlated feature functions, it avoids label bias problem and it lets us model the dependency between labels. Usually, linear-chain conditional random fields are defined on string sequence x with label y with a first order Markov assumption over the sequence

$$P(y|x,\lambda) = \frac{\exp \sum_{t=1}^{T} \sum_{s=1}^{S} \lambda_s \xi_s(y_{t-1}, y_t, x_t)}{\sum_y \exp \sum_{t=1}^{T} \sum_{s=1}^{S} \lambda_s \xi_s(y_{t-1}, y_t, x_t)}. \tag{3}$$

Variables x represent all sentence pairs in English–Lithuanian book corpus, i.e. (e, l). ξ_s are the feature functions that model relevant information about particular alignment position. Theoretically, it is possible to use alignment index a values as labels that are represented by variable y. However, such an approach would require much more data than is available for learning. Thus, as we already mentioned above, we use CRF model to label alignments that are generated from alignment generation processes based on maximum entropy models, sentence length and lexical properties. These labels are represented by variable y in Eq. (3).

The training of the mode (3) is done using two sets of labels. The first set represents labels that we already mentioned above:

- 'good' – as good alignment, i.e. other alignment processes can use this information as new alignment feature.
- 'bad' – bad alignment, i.e. other alignment processes can use this information to discourage these alignments to appear in the new alignment iteration.

We can try to use additional label set after we get $\hat{a}$ (Eq. (2)) from maximum entropy alignment algorithms and tag them with labels 'good' or 'bad'. These additional labels are:

- 'chapter' – it is a tag for some 'good labels and it is intended to find chapter names in a book.
- 'move left' – it is a tag for some 'bad' labels and it defines the suggestion to move target position to the left in order to improve alignments.
- 'move right' – it is a tag for some 'bad' labels and it defines the suggestion to move target position to the right in order to improve alignments.

Models like CRFs are appealing because they reduce each problem to a task of finding a feature set that satisfactory represents the problem at hand. Next, we describe the set of these feature functions that we used in order to label alignments.

1. The set of regular expressions that matches possible marks for book chapter (e.g. string pairs like 'IX'–'IX', 'Chapter 9'–'IX', etc.).

2. Paragraph indicators. We defined an indicator function that evaluates paragraph position likelihood.

3. Orthographic features. These are the features that measure how well sentences e and l match in terms of string overlap.

- Punctuation marks. We say that alignment that have matched punctuation marks like '?', '!', ')', etc., are more likely to be 'good'. These feature functions can be defined as $p(\textit{alignment label} = \text{'}\textit{good}\text{'} \mid \textit{matched puntuation mark})$.
- Capitalization. Feature function returns 1 if both sentences after alignment position a_i start with capital letters.
- Utterance unit marks. We add additional score to alignment if sentences after alignment position start with marks that represent sentences from utterance, e.g. '"' – marks some beginning of speech in English and '–' in Lithuanian.

4. Bilingual dictionary. Dictionaries are an important source of information about sentence alignment position. We set some predefined window of width 'w' (measured in number of words) around alignment position a and count how many words we can find in this window that are translations of each other when we look at the translation dictionary.

5. Named entities match. Named entity recognition component that indicates if there are named entities around alignment position a. We require that at least 3 char sub-string of a named entity must match in a language pair.

7. Phrase match. The same as words dictionary match. Usually we have a 'good' alignment if there is a long phrase match on both sides of translation.

8. Length based match. We increase alignment scoring if the length of sentences between sequential alignment points is similar.

5. CRF Model for Text Filtering

The purpose of the filtering algorithm is to find segments of a text in a book and to remove them if they are a part of book headers, page numbers, footnotes, etc. These formatting strings define the first type of strings that must be removed in order to get high quality alignments of bilingual books' corpus. Inaccuracies of the translation, i.e. a too figurative translation or just omissions of some sentences from a text define the second type of strings that we must consider for removal.

For the first type of strings we developed an algorithm that is based on a set of regular expressions that selects fragments of a text and marks them for removal. Table 1 presents a few examples of such regular expressions.

We tried several statistical models to define probability that a matched string with regular expression must be removed. Such strings as page headers can appear periodically. For this kind of strings the linear-chain CRF model can be used. On the other hand, we found that skip-chain CRF model (Sutton and McCallum, 2006) is better for labelling string sequences when we need to model long distance dependencies. For example, translation footnotes can appear only a few times in a book and the skip-chain CRF model can capture these long distance dependencies. Another example of long distance dependencies can be a chapter name in a book which is printed on each second page in the page

Table 1
Example set of regular expressions used to remove some fragments of text.

	Regular expression	Description
1	`^[ ]{0,2}(\d){1,3}[ \d]{0,3}$`	The line up to 8 char length. It consists only of digits and spaces
2	`(\d){1,3}[ \d]*$`	Select all digits at the end of the line (we found that sometimes PDF file converters add page numbers at the end of text line instead of inserting them into new line)
3	`(\p{L})\^\~(\p{L})`	Matches two letters that have '^~' string between them
4	`\r\n\r\n\r\n`	Three sequential strings of carriage return
4	`^.{1,30}[A-Z][A-Z].{1,30}$`	At least two UPPER letters and line of limited length
4	`^[\*].+$`	Select the line that starts with asterisk (useful for footnote detection)

header area. We would like to keep the chapter name at the beginning of a chapter but to remove all subsequent headers. Thus, the probability of a label sequence is modelled as

$$P(y|x,\lambda) = \frac{\exp\left(\sum_{t=1}^{T}\sum_{s=1}^{S}\lambda_s\xi_s(y_{t-1},y_t,x) + \sum_{u,v}^{T}\sum_{s=1}^{S}\lambda_s\xi_s(y_u,y_v,x)\right)}{\sum_y \exp\sum_{t=1}^{T}\sum_{s=1}^{S}\lambda_s\xi_s(y_{t-1},y_t,x)}. \quad (4)$$

We can see that the difference between a linear-chain CRF model and a skip-chain CRF model is that we use term $\sum_{u,v}^{T}\sum_{s=1}^{S}\lambda_s\xi_s(y_u,y_v,x))$ in a skip-chain CRF to model long-distance edges between text segments that were matched by the same regular expression. For filtering sequence y using skip-chain CRF model we define only three labels: 'keep', 'delete', 'review'. Labels 'keep' and 'delete' define recommendation either to delete a string or keep it in the text. Label 'review' means that only part of the string must be removed.

There are two stages of filtering process at which we try to detect these text segments. At the first stage we try to use only monolingual feature functions. If we fail to detect text segments that we want to remove at first stage then there is a possibility to detect them later in the second stage at which we use bilingual features (the same subset as for alignment algorithm described in the previous section). We used the following set of monolingual feature functions in our text removal process.

1. Probability $P(r_i)$. Each regular expression i has its own prior probability which can be interpreted as how probable it is that a matched string in a book must be deleted. As a simple example we can look at regular expression `^(\d){1,3}$`. It matches a line that has up to three digit numbers. Usually, such line will be a page number in a book, but occasionally it can be something else that we want to keep. Thus, initial probability that we want to delete content matched by this regular expression was set to 0.95. Coefficient λ_i for feature function that represents this regular expression was set to 0.05.

2. Indicator functions that mark if there are tokens that periodically appear in the text within the window of fixed width.

3. Line length. Indicator function that matches lines with a length shorter than twice compared with the average line length in the book.

4. Increase in the alignment probability 3 for the alignment label 'good' when we remove a matched text segment.

Finally, we present an algorithm that uses expression (4) and is capable of filtering out strings that can be interpreted as page formatting strings for bilingual book corpus alignment procedure. The input for the algorithm is a pair of books. The output is a set of strings that algorithm recommends to remove before the final alignment stage.

Algorithm 1 Filtering of noisy text segments

```
INPUT: A pair of books.
RETURN: A set of annotated strings that have been marked for deletion.
for each book do
    for each regular expression do
        Try match with RegEx.
        if Found then
            Put matched string and RegEx ID into U.
        end if
    end for
end for
for each s1 ∈ U and s2 ∈ U such that s1 ∩ s2 ≠ ∅ do
    add s1 ∪ s2 to U.
end for
for each s ∈ U do
    Compute alignment metric (3) with s and without s.
    Compute labels metric P(y|x, λ) using (4).
end for
```

The first loop of the algorithm iterates through each possible text segment matched by regular expressions and puts these segments into the set U. The second loop takes any two segments of the set U and creates a new string that is the product of intersection between these two segments. If this intersection is not empty then the new segment is put into the set U. After these two loops, in the set U we have all hypothesis for text filtering.

The last loop iterated through all the text segments s in the set U and calculates:

- Probabilities (3) of 'good' alignments with the text segment s removed and probability when it is not removed.
- These probabilities are used as feature functions in assigning labels to segment s using (4).

6. Proactive Learning

In the previous section we presented the algorithms for filtering and automatic alignment of translated books' corpora. There is always a possibility that filtering and alignment pro-

cesses will not achieve the required precision due to errors that appear during the translation and transformation of the original book. In order to correct these discrepancies we need to fully understand the world that is presented by the book author. Currently this full understanding of a text is beyond computer capabilities. Thus in our framework the alignment program can ask the reader for help by presenting several types of queries:

1. Alignment anchor point. The program can ask the reader to point two positions in a text: one in e and one in l that can be treated as an alignment.
2. Confirm filter decision to delete a sequence of text segments that have been matched by regular expressions.
3. Confirm filter decision to delete sentences from a book because they do not have translation equivalents.

The starting point for all these questions is how to formulate a comprehensive metric for query selection. One of the most common general methods for measuring informativeness of a query is information entropy. For a discrete random variable X, the information entropy is defined as: $H(X) = -\sum_i P(x_i) \log P(x_i)$.

We can use the entropy as an alignment informativeness $\phi(a)$ as follows. Let $\tilde{a}$ be the most informative alignment in the pool of all alignment hypothesis that we receive after we apply standard alignment procedure. We chose some alignment query strategy $\phi(a)$, which is a function used to evaluate each alignment a in the alignment hypothesis space A.

$$\phi(e,l) = -\sum_{\bar{a}} P(\bar{a}|e,l) \log P(\bar{a}|e,l), \tag{5}$$

where $\bar{a}$ ranges over all possible alignments between sentences in a book from a bilingual corpus.

Then, the most informative alignment $\tilde{a}$ will be expressed as:

$$\tilde{a} = \arg\max_{a \in A} \phi(e,l). \tag{6}$$

But for the alignment of books we must modify the query selection metric, which is based on entropy, in order to get better decision making process. Thus, to understand the problem with entropy as the metric of informativeness, we can imagine that we have a completely misaligned chapter between two books. It will be likely that maximum entropy of alignments will be somewhere in the middle of the chapter where we will have the highest uncertainty about alignments. Nevertheless, for the reader it will be difficult to point the 'good' alignment because that will certainly require for him to read through all the chapter in two languages. Then, we need to select queries that would be not just mostly informative but also convenient to for the reader.

There are few suggestions about possible modification in Settles and Craven (2008). In their study they have shown that in many situations one of the best metrics is information density metric.

$$\phi^{ID}(e,l) = \phi(e,l) \times \left(\frac{1}{U} \sum_{u=1}^{U} sim\left(el, el^{(u)}\right)^{\beta} \right). \tag{7}$$

Here the set U is a set of 'good' alignments in one page (excluding alignment el) where the page size is determined by user interface software and $el^{(u)}$ is some alignment from U. From Eq. (7) we see that the informativeness of alignment between e and l is weighted by its average similarity to all other 'good' alignments in U. Parameter β controls the relative importance of the density term. Similarity function $sim()$ is defined as cosine distance between two vectors:

$$sim(el, el^{(u)}) = \frac{\vec{el} \cdot \vec{el}^{(u)}}{\|\vec{el}\| \times \|\vec{el}^{(u)}\|}. \tag{8}$$

Vector el is defined from feature functions as a kernel vector:

$$\vec{el} = \left[\sum_{t=1}^{T} f_1(e, l, a_t, r_t), \ldots, \sum_{t=1}^{T} f_J(e, l, a_t, r_t) \right], \tag{9}$$

where $f_j(e, l, a_t, r_t)$ is the value of feature f_j for alignment a_t. e and l are a set of sentences in one page, which is presented in reader interface. Index t runs through all 'good' alignments in one page which we want to present for the reader and ask for some anchor point in alignment set or to confirm filter actions. T is the number of 'good' alignments in one page. The term r_t in Eq. (9) is a set of labels from the model that we presented in the section about CRF for sentence alignment.

Functions f_j in Eq. (9) can be similar to the features in CRF models that we used in sections above. What we found in this research was that it is possible to simplify this set of functions and have a measure of 'good' alignments density and to model some aspects of proactive learning. Then, in this research we chose the feature functions as follows:

1. Number of named entities that can be matched.
2. Probability of the phrase.
3. Number of matched infrequent words.
4. Number of question and exclamation marks.
5. Number of short utterance passages.

7. Evaluation

We begin our evaluation of this framework for books' corpus alignment by defining the alignment error rate (AER) (Och and Ney, 2003). Originally, AER was defined on a word-to-word level and it requires a manually aligned set of 'sure' (used for measuring recall) and 'possible' (used for measuring precision) links (referred as S and P). We suggest to redefine AER on a sentence-to-sentence level by defining the sets S and P as follows: a link $a_i \subseteq S$ if it links the beginning of sentences and $a_i \subseteq P$ if it links phrases in the middle of a sentence.

$$AER(A, P, S) = 1 - \frac{|P \cap A| + |S \cap A|}{|A| + |S|}.$$

Table 2
Corpus statistics for alignment quality assessment.

	Sentences	Words	Vocabulary	$\|S\|$	$\|P\|$
		Fiction books			
English	1675466	167914026	115847	1193836	2034015
Lithuanian	1668577	159958126	307710	1193836	2034015
		Non-fiction books			
English	78952	7912190	20157	58061	95610
Lithuanian	78668	7537294	53919	58061	95610

Table 3
Fiction and non-fiction books sentence-to-sentence alignment error rate for different methods.

Anchor p. Nr.	0	1	2	5	10	15	20	40
Alignment method				Fiction books				
hunalign	0.56	0.51	0.43	0.31	0.27	0.22	0.19	0.09
bleualign	0.55	0.50	0.31	0.30	0.26	0.21	0.19	0.09
bookalign	0.48	0.43	0.37	0.28	0.23	0.19	0.15	0.05
Alignment method				Non-fiction books				
hunalign	0.49	0.44	0.40	0.27	0.23	0.20	0.18	0.08
bleualign	0.46	0.42	0.40	0.25	0.23	0.20	0.17	0.08
bookalign	0.42	0.38	0.35	0.21	0.18	0.15	0.11	0.04

We evaluate our framework using two other methods against which we compare quality of corpus alignment. The first (*hunalign*) is the method suggested by Varga *et al.* (2007). Originally it was used for medium density languages like Hungarian, Romanian, and Slovenian. We chose it because we think that Lithuanian language can be described as medium density language. As a second method for estimating alignment quality we have implemented a method from Sennrich and Volk (2010) (*bleualign*). Because this method requires an automatic translation system, we used Google Translate system to translate Lithuanian language into English.

Table 2 shows the statistics of the corpus used to evaluate the method (we don't use morphology analysis when counting vocabulary words).

There were two questions that we considered, namely, whether the suggested sentence alignment method is useful for improving alignment precision of translated books and how alignment quality depends on a number of queries that a user must answer.

In order to answer these questions, we conducted the following experiment. We created the bilingual corpus of 200 books. Then we used all three methods (*hunalign*, *bleualign*, *bookalign*) to align this corpus. The error rate that we received after this step is shown in the first column of Table 3. Clearly, all three alignment methods scored bad. For example, error rate of *hunalign* method means that, on average, 56% alignments were erroneous.

After the first alignment iteration the system generated a query using the proactive learning approach described in this paper. Once the query has been answered, a new alignment iteration started. The second column in Table 3 shows error rates obtained after this step. We continued this alignment loop until we answered 40 queries.

It is clear from examining the results in Table 3 that all three methods improved performance after each answered query. What is particularly interesting, however, is that number of queries required to align books in order to get an error rate below 0.1 can differ significantly for each book. Nevertheless, the number of 40 queries appeared as a limit, after which all books can be aligned with acceptable quality.

8. Conclusions

We have suggested a model for the alignment of the bilingual English–Lithuanian books' corpus. In this research project we found that some translations appeared to be particularly difficult to align due to missing sentences or even paragraphs from the translated text. The important contribution of this study is that alignment accuracy was increased after we applied a new text filtering algorithm. This new filtering algorithm was developed using methods from previous studies in statistical machine translation and we have shown that its accuracy improves as new books are added to the bilingual corpus.

There are several factors that have a profound impact on alignment accuracy and that are impossible to sort out by fully automatic alignment procedure. Therefore, a new research is required to incorporate natural language processing and human computer interaction tools into bilingual corpus alignment system. One way to address these challenges is to use bilingual readers and integrate them with proactive learning framework. We have shown that alignments generated by currently available algorithms give errors that can be eliminated if a small number of proactive learning queries are used to filter text and create manual alignments.

In this paper, we have presented a solution of proactive learning for the task of aligning bilingual books. The proposed system is able to improve the alignment precision. Empirical results show that our method allows the alignment system to learn from each interaction with a human reader. We introduced several scenarios where users can choose between several options: confirm the suggestions from automatic alignment algorithm or enter alignments manually. Finally, it is worth to note that even if we tested the presented method on the English-Lithuanian language pair the ideas presented here can be used for other language pairs as well.

References

Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Net, H., Tomas, J., Vidal, E., Vilar, J.M. (2009). Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1), 3–28.

Berger, A.L., Della Pietra, V.J., Della Pietra, S.A. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), 39–72.

Braune, F., Fraser, A. (2010). Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 81–89.

Brown, P.F., Lai, J.C., Mercer, R.L. (1991). Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pp. 169–176.

Brown, P.F., Della Pietra, V.J., Della Pietra , S.A., Mercer, R.L. (1993). The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2), 263–311.

Chen, S.F. (1993). Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pp. 9–16.

Gale, W.A., Church, K.W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1), 75–102.

Lafferty, J., McCallum, A., Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the Eighteenth International Conference on Machine Learning, ICML*, pp. 282–289.

Laukaitis, A., Vasilecas, O., Laukaitis, R., Plikynas, D. (2011). Semi-automatic bilingual corpus creation with zero entropy alignments. *Informatica*, 22(2), 203–224.

Laukaitis, A., Vasilecas, O. (2008). Multi-alignment templates induction. *Informatica*, 19(4), 535–554.

McCallum, A., Nigam, K. (1998). Employing EM and pool-based active learning for text classification. In: *ICML*, Vol. 98, pp. 359–367.

Moore, R.C. (2002). Fast and accurate sentence alignment of bilingual corpora. In: *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas*, LNAI, Vol. 2499, pp. 135–144.

Och, F.J., Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*. 29(1), 19–51.

Sennrich, R., Volk, M. (2010). MT-based sentence alignment for OCR-generated parallel texts. In: *The Ninth Conference of the Association for Machine Translation in the Americas*.

Settles, B., Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 1070–1079.

Sutton, C., McCallum, A. (2006). An introduction to conditional random fields for relational learning. *Introduction to Statistical Relational Learning*, 93–128.

Thompson, C.A., Califf, M.E., Mooney, R.J. (1999). Active learning for natural language parsing and information extraction. In: *ICML*, pp. 406–414.

Tong, S., Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2, 45–66.

Varga, D., Halacsy, P., Kornai, A., Nagy, V., Nemeth, L., Tron, V. (2007). Parallel corpora for medium density languages. *Amsterdam Studies in the Theory and History of Linguistic Science*, 4(292), 247.

Xu, Y., Max, A., Yvon, F. (2015). Sentence alignment for literary texts. *LiLT (Linguistic Issues in Language Technology)*, 12.

A. Laukaitis has graduated from Vilnius University Faculty of Physics. He received the PhD degree from the Institute of Mathematics and Informatics, Vilnius. He is a professor of the Information Systems Department of Vilnius Gediminas Technical University. His research interests include text mining, natural language interfaces, machine translation systems and knowledge management.

D. Plikynas is affiliated as professor, senior research fellow at the Institute of Data Science and Digital Technologies in Vilnius University. He is also affiliated as professor, chief research fellow at the Department of Business Technologies in Vilnius Gediminas Technical University. He has been involved in a number of EU and nationally financed research projects. He has published 2 monographs, 8 chapters in books, over 40 publications and over 50 conference papers. His main field of interest includes fundamental and applied research (modelling and simulation), covering interdisciplinary research domains in natural and social sciences, e.g. computational intelligence, agent based simulations, complexity research, social networks, distributed cognition.

E. Ostasius is an associate professor of Vilnius Gediminas Technical University at the Faculty of Fundamental Sciences, Department of Information Technologies. He was awarded the candidate of mathematical sciences degree at Kaunas Polytechnic Institute in 1989, doctor of the mathematical sciences since 1993. His research interests include analysis of business processes and e-services, modelling, and evaluation in public and commercial sectors, their applications, and related issues.