UCL Université catholique de Louvain

Faculté de philosophie, arts et lettres (FIAL)

FIAL

Un éventail de formations, une multitude d'horizons

Automatic alignment of bilingual sentences

The case of English and Serbian

Mémoire réalisé par
Danica Seničić

Promoteur
Pr Cédrick Fairon

Année académique 2016-2017
Master en linguistique : finalité spécialisée traitement automatique du langage

Abstract

The aim of this thesis is to explore current systems for sentence alignment and adapt one for the automatic extraction and pairing of sentences in Serbian and English. For this purpose, the **EX**traction and **AL**ingment **P**ipeline was developed. `EXALP` takes unstructured data as an input, extracts the sentences, aligns them and presents them in readable format. Evaluated against manually extracted and aligned data, `EXALP` shows accuracy of 84%. The pipeline is easily adapted for other language pairs and its output can be applied in various domains of NLP and linguistics.

Abstract

L'objectif de cette thèse est d'explorer les systèmes actuels d'alignement des phrases et de les adapter à l'extraction automatique et à alignment des phrases en serbe et en anglais. Pour cela, l' **EX**traction et **AL**ingment **P**ipeline a été développé. `EXALP` prend des données non structurées en entrée, en extrait les phrases, les aligne et les présente en format lisible. Evalué contre les données extraites et alignées manuellement, `EXALP` montre une précision de 84%. Le pipeline est facilement adapté pour d'autres paires de langues et sa sortie peut être appliquée dans divers domaines du TAL et de la linguistique.

Dedicated to my father

Acknowledgements

I would first like to thank my thesis supervisor, pr Cédrick Fairon. He allowed this paper to be my own work, but steered me in the right direction whenever I needed it.
I would also like to thank my internship supervisor, pr Ranka Stanković from Belgrade University, who provided me with insight and resources concerning my thesis.
Finally, I must express my very profound gratitude to my parents, sister, my Belgian family, Matija, Ljiljana and my friends for providing me with unfailing support and continuous encouragement throughout my stay in Belgium and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

[1] I decided to include this section because it is closely related to the applications of parallel corpora and sentence aligners which are discussed in Chapter 5.

Chapter 1

Introduction

This thesis was written with the goal of expanding and obtaining a more in-depth picture of the work I did during my internship in the summer of 2016. The tasks I performed focused on the alignment of Serbian and English sentences for the purpose of creating a parallel corpus. More specifically, my research goal was to develop `EXALP`, the **EX**traction and **AL**ignment **P**ipeline, that will save time and effort in the future compilation of English-Serbian parallel corpus. This comprises the extraction of raw text, cleaning it up, extracting the sentences, aligning them and presenting them in a readable format. As a result, there are two main points to be stressed:

1. The evaluation of `EXALP` showed 84% accuracy measured against manually extracted and aligned data
2. `EXALP` is easily adaptable for most language pairs[1] with minor tweaks and adjustments

[1] Languages which are supported by the modules used in the pipeline

1.1 Outline of the thesis

The thesis has the following structure:

Chapter 1 will introduce the topic, as well as the current of state of research for Serbian in the domain of language technologies, and low-resource languages alike.

Chapter 2 is concerned with the internship and contains a detailed overview of the tasks I performed. This chapter is mostly self contained and serves to describe my experience and justify the choices I made in further research on the topic.

Chapter 3 presents the state-of-the-art research for sentence alignment. In the first part I make an overview of some of the available parallel corpora with the goal of examining the methods used in compiling them. Afterwards I present algorithms for sentence alignment and pick a few representative ones which I describe in more detail.

Chapter 4 is concerned with the project I did as a continuation of the work I did during my internship where I aimed to enhance and automatize the process of creating the parallel corpus. This part was a result of further research, improvement of programming skills and using various open-source resources. In the section that follows, I will evaluate how the program I developed performs against the manually aligned data I gathered during my internship.

Chapter 5 will take a look at possible applications such sentence aligners might have in various domains.

Chapter 6 will summarize the contributions of this thesis and lay out some suggestions for future research.

1.2 Sentence alignment

Parallel texts, or *bitexts*, represent a very valuable resource in many NLP applications, namely statistical machine translation [14], cross-language information retrieval [71] [33], word disambiguation [8], terminology extraction [37] etc. The key step in compiling parallel corpora is the alignment of parallel texts. The task is to determine which elements of the source text correspond to the element of the target text. The simplest level of alignment is *paragraph* alignment [56]. At this level, the pairing is relatively simple and paragraph aligners show good performance in most cases. The special case of paragraph alignment which may cause some issues is in the domain of literary texts [63]. The following step is the sentence alignment level. Parallel corpora on this level have shown the most utility in further applications [40]. Sentence alignment is the problem of finding correspondence between sentences in the source text and its translation in the target text, i.e. the nth segment of the source text and the nth segment of the target text should be mutual translations. At this level, the pairing becomes more complex and difficult. Certain sentences may be missing from either source or the target text and one sentence may correspond to two, three or more sentences in its counterpart text (see Fig. 1.1 and Fig. 1.2) [11]. Additionally, the translator may choose to change the structure of the text, the layout of the format or change the order of sentences which also makes the alignment more difficult.

The time consuming and labor intensive process of manually aligning the sentences makes the compilation of large parallel corpora difficult. Consequently, a number of automatic sentence alignment approaches are proposed [64] [22]. The automatic sentence aligners fall into three groups: the length-based, lexicon-based and those which combine the two.

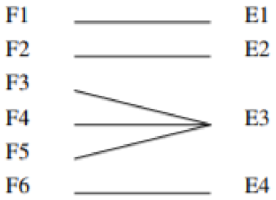


Figure 1.1: Example of the correct alignment between French and English sentences [11]

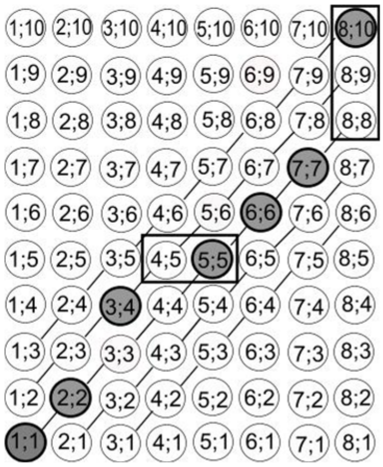


Figure 1.2: Example of alignment matrix. Every circle represents a pair of aligned sentences which are labeled with source sentence ID and target sentence ID. Bold circles stand for one-to-one alignments and bold squares for n-to-one and one-to-n alignments. Target sentence 3 remains unaligned. Target sentence 5 belongs to a one-to-two alignment. Source sentence 8 takes part in a one-to-three alignment. Filled circles denote model predictions [48].

The task becomes more complex with languages without enough lexical resources which consequently rely on heuristics and pure computational methods (e.g. character length, co-occurrences of words) and with *noisy* data. More precisely, there are data which contain the source text and its strict translation, and on the other hand, there are approximate translations[2] which contain a lot of noise. In reality, the second case is encountered more often. As stated in [40], when creating a very large corpus, the data can become very noisy, which means that there will be more zero-to-one and one-to-zero alignments than expected. For example, the compilation of the UN English-Chinese corpus [75] resulted in 6.4% of one-to-zero or zero-to-one alignments. Additionally, some data may be lost in the pre-processing steps, especially if the corpus consists of documents whose redaction is spread over a number of years. This usually results in various formats (`.pdf`, `.docx`, unknown formats etc.) and formatting. Moreover, layouts of texts in different languages often differ and change over time. The pre-processing steps include removing data such as tables, foot notes, end notes, references etc. Most of these steps introduce noise [40]. Because of the large number of documents, manually correcting it is impossible. Therefore, the sentence alignment needs to be "robust enough to detect noise and recover quickly if an error is made [40]". Following that line of thought, one should be careful when reviewing the evaluation tables of sentence aligners because they mostly work very well when clean data is introduced, but their performance drops with noisy inputs. The robustness of the aligners is even more important when dealing with low-resource languages, considering that most of the state-of-the-art aligners involve some kind of lexical input.

The final goal of a sentence aligner is to deliver high precision parallel sentence pairs which

[2] During my internship I worked on a set of data which consisted of scientific journals in Serbian and corresponding editions in English. However, even though they supposedly had the same content, there were data that occurred only in one of the editions, like metadata or claims that needed additional explanation in English. In such cases those sentences would not have their respective pair, but would be removed from the alignment process.

will fuel machine translation systems or create translation memories for specialized domain. In the cases of these applications, it is safe to discard sentence pairs with low level certainty. The research thus far showed that keeping only one-to-one sentence alignments makes these systems prone to literal translations which is why more interest is raised into finer-grained alignments [73].

Alternatively, the condition of keeping the entirety of the text and its respective translation may be required for more in-depth analysis. These are mostly of linguistic nature, like analyzing language change, translation mistakes, translation evaluation, examining samples of learners' language. Additionally, it may be used for the compilation of future corpora. In other words, such alignment would need to result in 100% accuracy. This remains a very complex task to automate and the solution would be to begin with the sentence aligner and subsequently correct the output manually.

1.3 Serbian language in the digital age

It is estimated that there is between six and seven thousand languages currently spoken in the world [24], however most of the research is focused on a very small number of languages. Languages that receive most of the attention in the NLP research usually have plenty of tools and resources due to social, political and financial motivation. Gathering resources is one of the first steps that leads to in-depth NLP applications. However, due to lack of funding most languages remain in the early research phase. Furthermore, statistical approaches and neural network training have become most of the state-of-the-art methods in NLP research [60], leaving no room for using any rule-based or non-statistical methods. Consequently, most of the research heavily relies on huge amounts of data which are scarce in the case of low-resource languages. Without such data, any statistical approach will give subpar results.

Serbian can be considered a low-resource language in terms of tools and methodology. There are large amounts of unstructured data which would greatly benefit from the development of such tools and consequently enlarge the crucial resources.

The contemporary Serbian language is one of the standardized varieties of the Serbo-Croatian which was in use until 1991 [25]. The authors in [66] point out certain properties of Serbian which need to be taken into account from the computational point of view:

- The use of two alphabets

  Serbian language is one of the rare examples of a synchronic diagraphia. Latin and Cyrillic alphabet are equally used and speakers of Serbian can read and write both scripts. There are a number of tools which allow for successful transliteration between the scripts.

- Phonologically based orthography

  Consequence of this property is that all specificities of different varieties of Serbian can be seen in the written form (ml**e**ko, s**e**ćanje/ml**ij**eko, s**je**ćanje[3]), as well as all types of morphophonemic changes (vra**b**ac/vra**p**ci[4]). Both are very frequent.

- Rich morphological system

  Reflected on both derivational and inflectional level.

- Free word order

  Subject, predicate, object and other constituents have free word order.

[3] Varieties of the words *milk* and *memory* in Ekavian and Ijekavian dialect.

[4] Assimilation of consonants: **b** is transformed to **p** in the example of *vrabac* (sparrow, sg.) and *vrapci* (sparrows, pl.)

- Complex agreement system

  For example, there is agreement between adjectives and nouns in gender, number, case and animacy.

The META-NET [6] Network conducted a study on European languages and the support they receive through language technologies. The material is presented in *white papers* which are the overviews of each language, written in source language and aligned with its English translation.

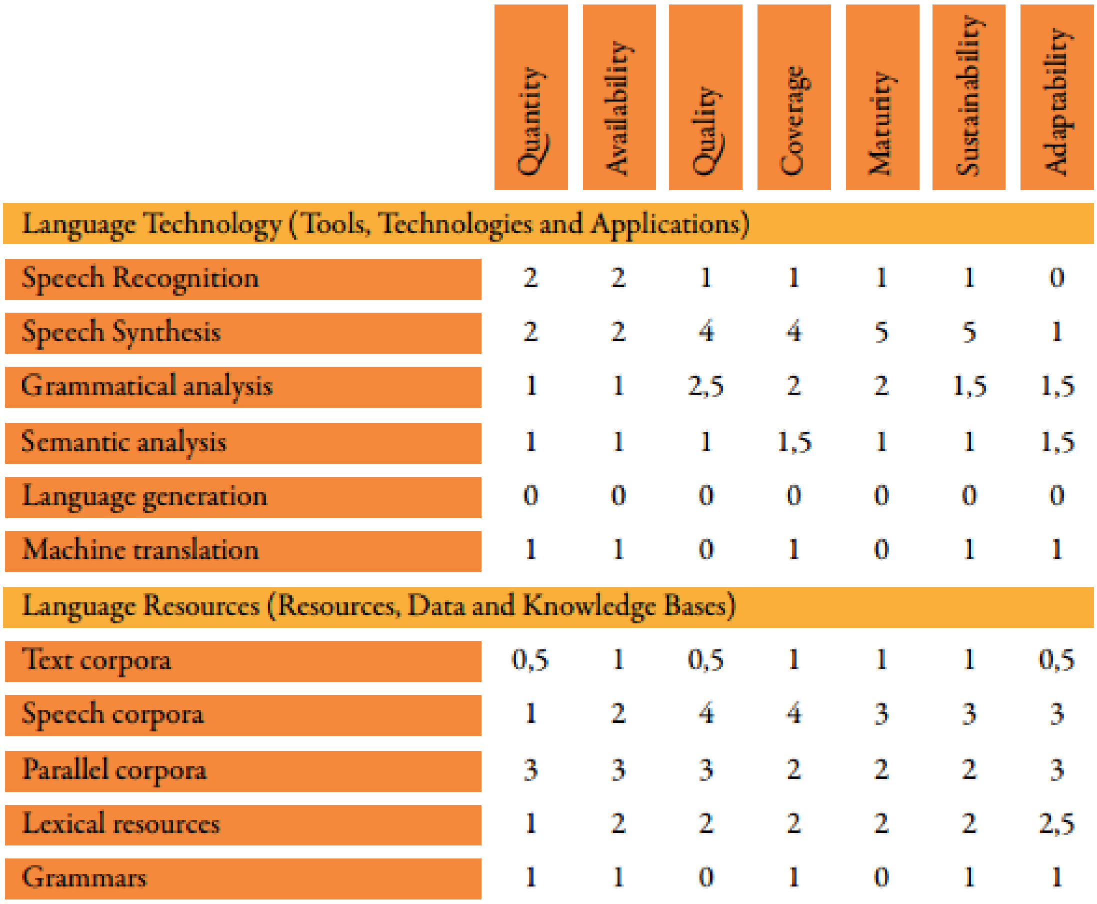
The Serbian edition of *white paper* [69] states that the industry of language technologies is relatively underdeveloped compared to the leading EU economies. One of the primary reasons of such a state is that there is no national programme which would support this kind of activity, therefore the development and application of these technologies is often unsynchronized. Only recently had the Ministry of Education and Science recognized the interdisciplinarity in this field, which had previously been strictly divided between mathematics, computer science and linguistics, and funded a series of projects in the period from 2011 to 2014. These projects allowed for the completion of corpus of contemporary Serbian language which was compiling in the period from 2002 to 2014. This was an important step as it enabled the development of electronic dictionary of simple words and the initiation of the dictionary of compounds. Additionally, aligned French-Serbian [67] [68] and English-Serbian[5] [34] corpora of literary texts were compiled. An important point was the development of software tools such as LeXimir [5] (a workstation which enables integration and transformation of heterogeneous lexical resources) or ACIDE [63] (integrated development environment for aligned corpora) which I had the opportunity to work with during my internship.

[5] Described in more detail in the section of currently available parallel corpora and MULTEXT-East project

In regards to morphology, the level of the technologies and resources is satisfactory, mostly due to the existence of monolingual electronic dictionaries and local grammars. Consequently, tools for information retrieval and extraction are available, as well. However, bilingual dictionaries in machine readable format are still hard to get by. Speech processing is being realized at the University of Novi Sad. This is a branch mostly developed by engineers and is considered to be well developed, but with the need for extensive research and further applications. The rest of the applications, like shallow parsing, summarization, machine translation, ontological resources, remain in experimental and research environments.

With the complexity of the Serbian syntax on one hand and the lack of initiative in research on the other, all areas based on deep parsing structures do not exist. Those are branches like sentence semantics, text semantics and language generation. This also results in the absence of a formalized syntax for Serbian language and restricts the development of syntactically and semantically annotated corpora. Thus, the formalization of Serbian syntax is one of the most urgent tasks to be accomplished. This was stated as a significant gap in the development of the language technologies in Serbia in 2012 when the *white paper* was published, and still there has been no significant advance in this field.

To conclude this part, Fig. 1.3, taken from the *white paper*, gives an overview of the state-of-the-art of language technologies and resources in Serbia in 2012. It should be noted that 6 is the highest and 0 is the lowest score.



	Quantity	Availability	Quality	Coverage	Maturity	Sustainability	Adaptability
Language Technology (Tools, Technologies and Applications)							
Speech Recognition	2	2	1	1	1	1	0
Speech Synthesis	2	2	4	4	5	5	1
Grammatical analysis	1	1	2,5	2	2	1,5	1,5
Semantic analysis	1	1	1	1,5	1	1	1,5
Language generation	0	0	0	0	0	0	0
Machine translation	1	1	0	1	0	1	1
Language Resources (Resources, Data and Knowledge Bases)							
Text corpora	0,5	1	0,5	1	1	1	0,5
Speech corpora	1	2	4	4	3	3	3
Parallel corpora	3	3	3	2	2	2	3
Lexical resources	1	2	2	2	2	2	2,5
Grammars	1	1	0	1	0	1	1

Figure 1.3: State of language technology support for Serbian [69]

Chapter 2

The internship

During the period from June to October 2016 I did an internship within the Society for Language Resources and Technologies (Društvo za jezičke resurse i tehnologije)[1] in Belgrade, Serbia. I was supervised by professor Ranka Stanković who is also one of the founders of the aforementioned society. During my internship I did some of the work part-time, with a summer break when the Institute was closed (July 2016), where the overall contribution sums up to 210 hours of work.

Part of my master's program in Natural Language Processing (NLP) was to do an internship during my two years of study. I wanted to do the internship in Serbia because that is where I intend to seek employment once I have obtained my master's degree, therefore I wanted to get myself familiar with the state-of-the-art of NLP in this region.

In the following section I am going to summarize and explain the tasks I had during my internship, with comments and additional information which are the result of my background research.

[1] `http://jerteh.rs/` Accessed July 27 2017

2.1 Society for Language Resources and Technologies

This society was founded with the objectives of popularizing and promoting all branches of linguistic technologies, at a scientific, practical and professional level. It has participated in and accomplished a number of projects, independently or with the support from external projects e.g. PARSEME and COST. Other than that, this society organizes meetings, seminars, conferences, lectures, panel discussions, workshops and similar events. Furthermore, there is the publication of books, manuals and newsletters. This society was founded with the goal of exchanging opinions and sharing advice among its members, while cooperating with educational and research institutions, associations, public authorities, companies and other entities in Serbia and abroad.

It is the first and only organization in Serbia which promotes the advances in computational linguistics and works on the digitization of Serbian language [4].

2.2 Detailed overview of the tasks

In this section I will explain the tasks I did during my internship. I spent most of my time on the bilingual corpora alignment. I did the work either in the offices at the Faculty of Geology and Mining in Belgrade, where my internship supervisor teaches, or I did it from home with regular communication via e-mail.

2.2.1 Bilingual corpora alignment

The first and the task I spent most of my time with was aligning corpora compiled out of English and Serbian texts. These texts were later incorporated in the existing collection of aligned parallel texts which make up a bilingual digital library in a tool called Bibliša.

Bibliša

Bibliša is a tool developed within the Society for Language Resources and Technologies at the University of Belgrade. Its goal is to enhance search possibilities in bilingual digital libraries and offer contextual translation. The tool offers cross-lingual search functions for large collections of texts. The user may use both simple and multiword keywords in more than one language. The search is based on keywords with additional use of other resources, like e-dictionaries, semantic networks and termbases. This allows for searching by concept. Additionally, Bibliša relies on tools that allow for the expansion of user queries both semantically and morphologically, which is very important in a highly inflective language such as Serbian. Finally, the tool allows the full-text and metadata search, with the concordance sentence pairs for translation and terminology work support [55] [1].

Step by step process to aligned corpora

Roughly the task of creating aligned parallel corpora consisted of the following steps:

1. Text extraction

   All the texts I worked with were initially found in the `.pdf` format. This is because all the texts were scientific journals and could only be found in the processed format. They either appeared in two different `.pdf` files (*Management* and *Serbian Dental Journal*) or in the same file (*The Serbian Language in the Digital Age*). There were three main hurdles to overcome:

- Conversion to `.txt` file

  This was done by either using software like ABBYY PDF Transformer[2] for `.pdf` to `.txt` conversion or, when this kind of conversion would result in too many mistakes, the file would get exported to `.docx` file and would then be touched up manually. The problem that often occurred was that Serbian letters with diacritic marks (š,đ,č,ć,ž) were not correctly converted. Since this is a very recurring issue when working with Serbian text files, my supervisor already had a tool that converted it to a correct `.txt` file and had provided us with the `.pdf` files converted to `.txt`.

- Cleaning up the `.txt` file

  The converted `.txt` file had a lot of noise that was not needed in the parallel corpora material, such as: information in the headers and footers, page number, references, reference numbers, image captions, tables, table contents etc. This is not an easy task to be automated mostly because each journal has a different format. Each article from the journal was saved separately. This part was done manually.

- Verification of the extracted text

  Once the text has been extracted, there are two files of supposedly same content. They need to be skimmed through to see if that is the case. The errors that may occur here are those made because of the difference in the layout, so a part of sentence may be lost or found somewhere else in the file. It may also occur that some sentences are excluded, i.e. they are not translated or they have inadequate translation. In such cases unpaired or incomplete sentences are deleted. Additionally, the paragraphs were separated by one carriage return and line feed.

[2] `https://www.abbyy.com/en-eu/pdf-transformer`

2. Creating an XML document

   XML or eXtensible Markup Language is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable [3]. The idea behind the Text Markup is to enrich plain text with additional information for document structure, meta-information (authors, version), linguistic annotation and layout information. It has found its applications in NLP and can be used for tokenizaton, morphological analysis, part-of-speech tagging, chunking, named entity recognition (NER) and sentence boundary recognition [53].

   The structure of each XML file must be well-defined and contains elements such as:

   - Tags, that may have attributes with associated values

     ```
     <tag>
     <element attribute = "value">
     ```

   - Text content

     ```
     text content
     ```

   - Comments

     ```
     <!--comment-->
     ```

   - Entity references

     ```
     &title;
     ```

   - Other elements, e.g. for encoding and namespace information, document type declarations, processing instructions

   Furthermore, each XML document must have at least one element and exactly one root element. The elements may embed other elements and form a tree structure. Start and end tags, i.e. opening and closing tags must balanced. Certain elements are predefined : `&, <, >` and should be omitted from the content, i.e. the document may contain only legal characters.

First line in the XML document is a declaration that identifies the document as being XML. All documents should begin with such declaration:

`<?xml version="1.0" encoding="UTF-8"?>`

where `version="1.0"` is the version of the XML standard that the XML document conforms to. Every XML processor is obliged to support Unicode. Most other encodings are optional, i.e. they depend on the implementation. Correct encoding is especially important if we intend to work with multilingual documents. To sum up, XML documents must be well-formed, otherwise they report an error and are not valid. An optional step which is often a part of the XML document is a validation scheme such as `DTD` (Document Type Definition). A `DTD` defines admissible and required elements, elements nesting, sequence choice, optionality, required and optional attributes and their default values.

- Preparation of XML files for Bibliša

  The XML files I worked with needed to have the following structure:

  - `<?xml version="1.0" encoding="UTF-8"?>` declaration
  - predefined `DTD` : `<!DOCTYPE body SYSTEM "body.dtd">`

    The `body.dtd` file has the following content:

    ```
    <!ELEMENT body (div+)>
    ```

```
<!ELEMENT div (head | p)*>
<!ELEMENT head (#PCDATA)>
<!ELEMENT p (seg)+>
<!ELEMENT seg (#PCDATA)>
```

which means that every XML document which is associated with it needs to have the following structure:

```
<body>
<div>
<head>Document_title</head>
<p><seg>Example text</seg>
<seg>Example text</seg></p>
<p><seg>Example text</seg></p>
</div>
</body>
```

If one should try to save an XML document with a structure which does not correspond to a DTD document it is associated with, an error from the XML validator will occur. The validator I used was a plugin for Notepad++ XML Tools [7].

- <seg> and <p> tags where <seg> corresponds to a segment, i.e. a sentence, and <p> to a paragraph

In order to tag the segments appropriately with the <seg> tags, it is important to extract them as precisely as possible. The segments include sentences, titles, subtitles and enumerated items. This was done by using Unitex [49]. Unitex offers support for both English and Serbian. With the option of preprocessing the `.txt` file, in the output we get the file with the `.snt` extension, where at the end of each segment there is a `{S}` string. For example:

```
Keywords:  Panel data, DEA, Window analysis, Banks efficiency,
Serbian banking{S} 1.  Introduction{S} In this paper we analyze
the performance of twenty-eight commercial banks in Serbia
over THE period 2005 - 2011.{S} Available data before 2005
are not comparable because in Serbian banks' reporting was
not regulated by the law.{S} In 2005, as the independent
institution, the National Bank of Serbia implemented the
regulations for banking system in Serbia.{S} The evolving process
of the banking system started in 2001 along with the transition
of the Serbian economy when the country had approximately 90
banks.{S} Since that year, until now, some banks were liquidated.{S
Some merged with others, and the remainder were privatized.{S}
```

The same was done for both Serbian and English files. Accordingly, the Serbian equivalent is:

```
Ključne reči:  Panel podaci, DEA, Window analiza, efikasnost
banaka, srpsko bankarstvo{S} 1.  Uvod{S} U ovom radu će biti
analizirane performanse dvadeset osam komercijalnih banaka
u Srbiji u periodu od 2005-2011.  godine.{S} Dostupni podaci
```

```
pre 2005.  godine nisu uporedivi zato što izveštaji srpskih
banaka nisu bili zakonski regulisani.{S} Nezavisna institucija
"Nacionalna Banka Srbije", 2005.  godine donosi propise u
bankarskom sistemu Srbije.{S} Razvojni proses bankarskog sistema
započet je 2001.  godine zajedno sa tranzicijom srpske ekonomije,
u trenutku kada je država imala približno 90 banaka.{S} Od
te godine pa do danas, neke banke su likvidirane.{S} Neke su
pripojene ostalima, a preostale su privatizovane.{S}
```

After all the `.txt` files have been preprocessed with `Unitex`, the `{S}` strings need to be replaced with `<seg>` tags and taking carriage returns and line feeds into account add the `<p>` tags, as well.
In the beginning this was done manually using the `Find` and `Replace` option in Notepad++. The replacement was done in three steps:

(a) `{S}` replaced by `</seg>\r\n<seg>`

(b) `<seg>\r\n` replaced by `\r\n<p><seg>`

(c) `</seg>\r\n` replaced by `</seg></p>\r\n`

After these replacements, the first opening tags and last closing tags were missing, so they needed to be added manually, as well as the rest of the needed XML structure (declaration, `<head>, <body>, <div>`).
After I had done a few entries following these steps, I realized this was a repetitive task that in overall took up a considerable amount of time, but one which could

be automated. I made a script in Perl `unitex_snt2xml.pl` which takes in its argument the preprocessed `.snt` file and gives an `.xml` file which corresponds to the predefined DTD structure, as its output. The name of the file is used to fill in the `<head>` tag and by using regular expressions the `{S}` conversion to `<seg>` and `<p>` tags has been made. Additionally, some special cases, like double editions of the journals, have been handled.

Finally, once I had the proper XML files, I needed to make sure that Serbian and English file contained the same number of segments which should later be aligned, and ideally segments that correspond to each other. This is the part that is the most consuming and takes a reasonable amount of human input which does not have an obvious solution to automatization. The issues that needed to be corrected were mostly the following:

- Two sentences in one segment, mostly due to typing errors (e.g. two spaces after the end of sentence, instead of one; lowercase letter at the beginning of the sentence)
- The segment does not exist in one of the two languages: it was either lost in the preprocessing or the translator decided to omit it.
  This was corrected by deleting the sentence that had no pair.
- The segments are misplaced, i.e. they exist in both languages, but are found in different places in the text.
  This was corrected by consulting the original file and putting the segments in order.
- One sentence is translated by two or more sentences, i.e. two or more sen-

tences are translated by one sentence.

This was corrected by cutting the sentence into smaller parts, as it is preferable to have more small segments than less big segments. This error has no pattern, as the number of sentences depends on the translator's judgment. The recurring one that I noticed was that Serbian had longer sentences, with one or more relative clauses, whereas in English this would rather be translated by putting each clause in a separate sentence.

Once the XML file has been validated and the English and Serbian one contained the same number of segments, they were ready for the next step.

3. TMX

TMX, more precisely Translation Memory eXchange, is an XML specification for the exchange of translation memory data. It is often used in tools for computer-aided translation (CAT). This is an excerpt of one TMX file:

```
<tu>
      <prop type="Domain">Savic et al., 2012, No. 65,
      ID: 7.2012.65.1</prop>
      <tuv xml:lang="en" creationid="n21␣"
      creationdate="20160926T220611Z">
        <seg>Among the banks in Serbia (33 banks),
        we can find a number of
        banks that are still (or at least partially)
        owned by the Republic of Serbia (8 banks). </seg>
      </tuv>
```

```
        <tuv xml:lang="sr" creationid="n21␣"
        creationdate="20160926T220611Z">
          <seg>Medu bankama u Srbiji (33 banke)
          mozemo uociti veliki broj
          banaka koje su i dalje (ili su bar delimicno)
          u posedu Republike Srbije (8 banaka). </seg>
        </tuv>
  </tu>
  <tu>
        <prop type="Domain">Savic et al., 2012, No. 65,
        ID: 7.2012.65.1</prop>
        <tuv xml:lang="en" creationid="n22␣"
        creationdate="20160926T220611Z">
          <seg>Some of the banks are foreign banks (21 banks)
          and there are also public-privately owned banks
          (4 banks) (NBS | Banking Sector, 2012). </seg>
        </tuv>
        <tuv xml:lang="sr" creationid="n22␣"
        creationdate="20160926T220611Z">
          <seg>Neke od njih su inostrane banke (21 banka)
          a ima i nekoliko njih koje su domace privatizovane
          (4 banke) (NBS | Banking Sector, 2012). </seg>
        </tuv>
  </tu>
```

We can see that at the beginning of each pair there is a specification of the article: author, year, journal number and ID. Additionally, each pair has its own creation id (in this case n21 and n22) which allows for their alignment.

The TMX is created by ACIDE - an integrated environment for the preparation of aligned corpora. This environment provides graphical user interface for alignment and visualization of aligned text. It has more functions available, but those are the ones I had used. ACIDE performs alignment using the program packages XAlign and Concordancier which were developed at the LORIA Laboratory in France [63].

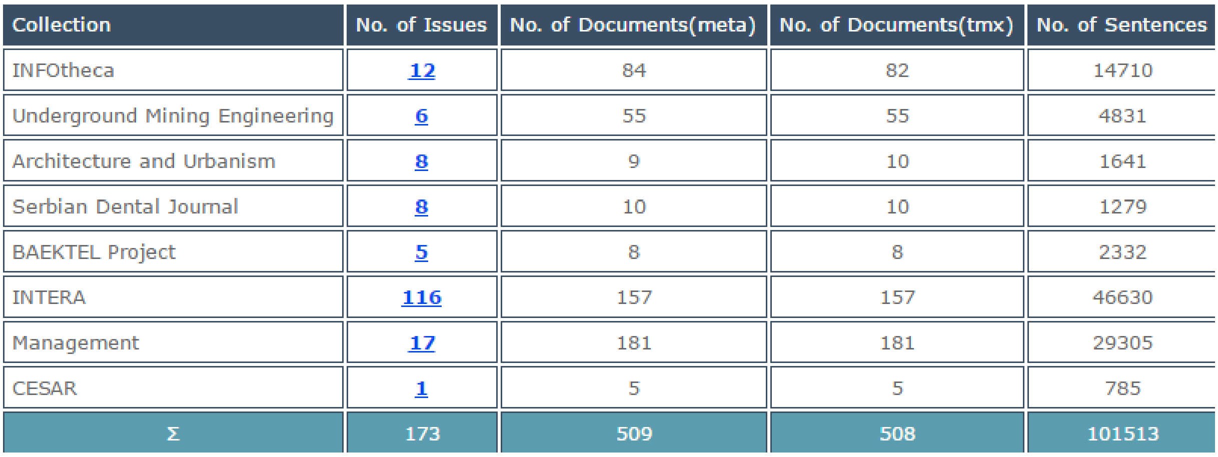
In ACIDE, after opening a new project, I would upload the XML file in English and a corresponding XML in Serbian. Afterwards I would type in the needed metadata (author of the article, volume and number of the journal, year of publishing). If the pairing was done successfully, I would have access to a TMX file, as the one given as an example above. However, if the files did not contain the same number of segments, I would get an error and would have to review my XML files again.

4. Uploading the files to Bibliša

   Once I had valid TMX files, I would send all files I worked with to my supervisor and she would upload the TMX files to Bibliša.

Conclusion and overall results

During my internship I aligned 31369 sentences, out of which 29305 are from the journal *Management*, 1279 from the *Serbian Dental Journal* and 785 from *CESAR*. This was a considerable addition, as the database now has 101513 aligned sentences (Fig. 2.2, Fig. 2.1).



Collection	No. of Issues	No. of Documents(meta)	No. of Documents(tmx)	No. of Sentences
INFOtheca	12	84	82	14710
Underground Mining Engineering	6	55	55	4831
Architecture and Urbanism	8	9	10	1641
Serbian Dental Journal	8	10	10	1279
BAEKTEL Project	5	8	8	2332
INTERA	116	157	157	46630
Management	17	181	181	29305
CESAR	1	5	5	785
Σ	173	509	508	101513

Figure 2.1: Screenshot of the statistics in Bibliša

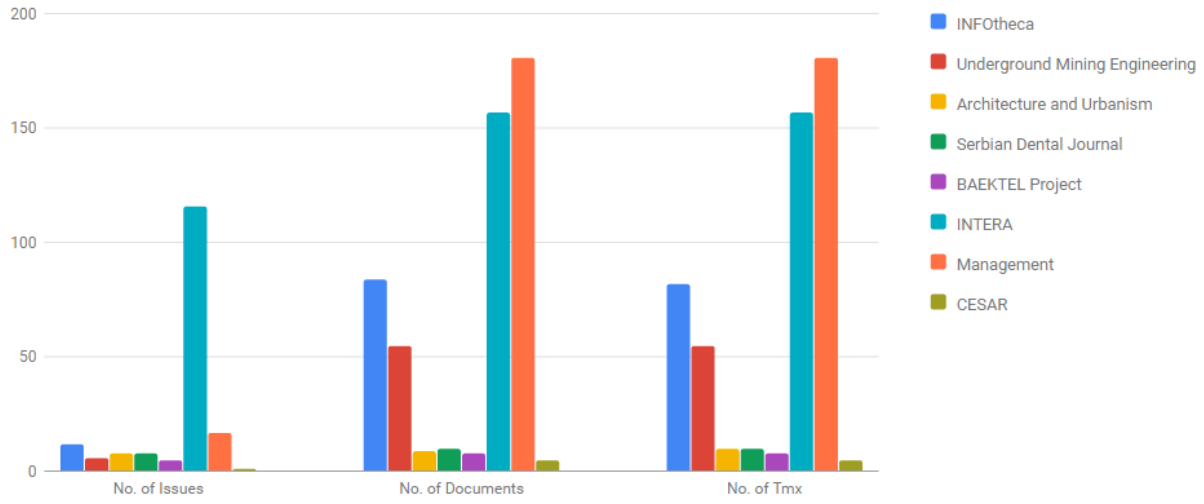


Figure 2.2: Statistics of the collections found in Bibliša

This task was the central task of my internship. I learned about the corpora alignment and what are its main challenges. As for my contribution, I believe that the script I made saved a considerable amount of time in this task. Additionally, it allowed for the uniformity of the files, which later made them easier to read. Its usage is very straightforward and can be used in the future by anyone who participates in this task. However, there are other features I believe in the long run could be added in the script, as well, like integrating it with `Unitex` and allowing it to process multiple files.

2.2.2 Terminology extraction [3]

LeXimir is a toolkit for development and management of lexical resources developed by the Society for Human Languages and Resources. One of the many tasks preformed by LeXimir is the extraction, evaluation and description of terminology. The extraction is

[3] I decided to include this section because it is closely related to the applications of parallel corpora and sentence aligners which are discussed in Chapter 5.

done via the automatic procedure for single and multiword units extraction and lemmatization. After they have been extracted, they are analyzed and compared to the words that can already be found in the existing terminological dictionaries. Words which are not found in the dictionaries are candidates to be included. My task was to analyze the Excel sheet where I needed to determine if the extracted word was properly lemmatized, if it was a collocation or a compound and if it belonged to terminological lexicon of a certain domain. The task consisted of filling in the empty cells in the Excel sheet (1 for true and 0 for false). More precisely, the sheets I worked with were extracted from the journal *Managament* I have previously been aligning.

2.3 Conclusion

During my internship I learned about the state of the Serbian language in the digital age, what are the most urgent tasks to be accomplished and what has been done so far. I was thoroughly introduced with the concept of aligned corpus, its applications, process of compilation and its main challenges. The time I spent with it is an important input for the work I intend to do for my master thesis, as the most efficient task automatization can be done by someone who has done the work manually and is aware of all the details of the process.

The competences and skills I have acquired during my first year at UCL allowed me to handle these tasks more efficiently in many ways. The programming skills I acquired allowed me to handle my tasks more efficiently and make a contribution.

Chapter 3

State of the art

In this chapter I will introduce the concept of a corpus in the context of computational linguistics. Exploiting well designed, representative corpora is one of the main pillars of successful natural language processing, therefore it is important that their compilation is reliable and of good quality. However, since this thesis is concerned with a specific issue in a very large domain of corpus linguistics, the overview of corpora in general will be discussed very briefly. In the introducing section I will mention important corpora through the history, but the further focus will be on the parallel corpora. In the sections that follow, I will narrow down to the topic of the alignment of sentences for parallel corpora. Finally, I will discuss the current algorithms used in this domain.

3.1 Introduction

In the domain of computational linguistics, we can define a corpus as a "body of linguistic data, usually naturally occurring data in machine readable form, especially one that has been gathered according to some principled sampling method [46]." In other words, a corpus is a large body of text which is suitable to be computationally processed, is adequately annotated and is a representative language sample. The collection of such corpora was virtually impossible during the pre-computer age, therefore the first compilation of a corpus for such purposes dates back to the 1940s and the work of Roberto Busa [13]. Busa was involved in research concerning the aquinian philosophy and wanted to gather all the sentences in the oeuvre of Thomas Aquinas which contained the preposition *in*. The painstaking work of thirty years ensued, along with numerous participants on the project which resulted in a digitized version of 56 volumes of *Index Thomisticus* published on a CD-ROM in 1989. However, even at this rudimentary level it was important to go through the steps of pre-processing, lemmatization and morphological annotation [13] which remains at the core of any corpora compilation today.

Further on, the pioneering work in English corpus linguistics was done by Francis et al. in 1979 with the Brown Corpus [21] which contains roughly one million words of American English from a wide variety of sources. In the years to come, the number of available corpora became larger and the amount of data steadily increased [46]. Worth mentioning is the British National Corpus (BNC) [36] published in 1994 which already contained 100 times more words than the Brown Corpus. Some further references as an insight into the evolution of linguistic corpora are Survey of English Usage [51], corpus of transcribed spoken language [58], part-of-speech tagged corpora [21] [23].

However, since the work I have been focused on is mainly concerned with parallel corpora, I would like to narrow down this chapter to this particular type of corpus, after introducing the basic division of existing corpora.

3.2 Parallel corpora

The corpora mentioned thus far all belong to a group of *monolingual* corpora, meaning that they contain text of a single language, while *multilingual* corpora contain texts of two or more languages. The collection of various monolingual corpora of different languages that share the same topic are called *comparable* corpora. A corpus that contains a direct translation of the text into one or more languages is called a *parallel* corpus or a *bitext*. It may seem that parallel and comparable corpora contain roughly the same data, but their difference is significant [46]; if the goal of the research is exploring the contrastive differences between the languages, it may be advisable to go for the comparable corpus, as the subtle nuances between the languages may be lost in the translation process. However, for the purposes of applications such as statistical machine translation, translation by examples, terminology mining etc., parallel corpora make an excellent resource.

Going through the history of parallel texts, it would be impossible not to mention the famous Rosetta stone which dates back to 196 BC and was discovered in 1799. The most special thing about this ancient stone was not that it represented content in the source language and its translation; throughout the history it was not rare to encounter multilingual versions of contracts, treaties, sacred texts, literature etc. However, the particularity of this artefact is that it contains three different inscriptions of the same text and on the same medium (the top and middle text are in Ancient Egyptian, using hieroglyphic and Demiotic script, and the bottom one is written in the Ancient Greek) [65]. The discovery of this stone allowed for deciphering of ancient hieroglyphs and today, according to the Merriam Webster dictionary [70] the idiomatic expression *Rosetta stone* means "one that gives a clue to understanding". In other words, it is a crucial key "in the process of decryption of encoded information, especially when a small but representative sample is recognised as the clue to understanding a larger whole[1]".

On the other hand, the first attempts at using parallel texts in the automatic processing were noted in the 1950s [32] with the goal of machine translation, however limited storage, low computational capacities and difficulty of entering large quantities of data probably account for restricted use of corpora at this time [65]. In the 1980s the compilation of aligned, parallel texts continued. Once again, resources of stored sample translations were intended for the purposes of machine translation. The alignment was proposed on several levels: paragraphs, sentences, expressions and words. Later on, the applications of aligned texts became more diverse: the compilation of translation memories, derivation of bilingual dictionaries, terminology mining, cross-language information retrieval etc [65]. With the growth of storage space and computational resources, as well as the development of the Word Wide Web, the parallel corpora became more accessible and easier to compile. In the following section I will list some of the most famous available parallel copora.

3.3 Currently available parallel corpora resources

The source for parallel texts are often multinational institutions because of the documentation they create through their activities. Such institutions are United Nations or the European Union, but also governments of bilingual or multilingual countries such as Belgium (French and Dutch), Canada (French and English) or the Hong Kong (English and Chinese) [29]. The ever growing collection of these documents and the improvement of the resources especially showed promising results in statistical machine translation [29]. This can be noticed in practice, when using translation engines such as Google Translate, the

[1] *Oxford English Dictionary* (1989) s.v. Rosetta Stone Archived June 20, 2011 at Archive.is; retrieved from en.wikipedia.org/wiki/Rosetta_Stone

pairings of the languages with rich resources will show significantly better performance than low-resource languages.

In this section I will enumerate some of the most popular parallel corpora and explore the methods which were used in obtaining them.

3.3.1 Hansards

`https://www.isi.edu/natural-language/download/hansard/`

This corpus contains 1.3 million pairs of aligned text chunks in French and English from the official records (*Hansards*) of the 36th Canadian parliament. The initial records in HTML format were made available by the Canadian government and the raw text was extracted with a Perl script *HTML::Parser*[2]. Sentences were extracted by using a maximum entropy classifier for sentence boundary detection, *mxTerminator*[3][52]. The alignment of the sentences was achieved with *GSA Tool*[44] which uses Smooth Injective Map Recognizer, a parallel text mapping algorithm. The *Hansards* corpus is often used for solving machine translation problems.

Some caveats: the corpus is compiled of proceedings of the Canadian parliament, therefore the content is limited to legislative discourse.

[2] Available at `http://search.cpan.org/~gaas/HTML-Parser-3.72/Parser.pm`, developed by Gisle Aas and Michael A. Chase

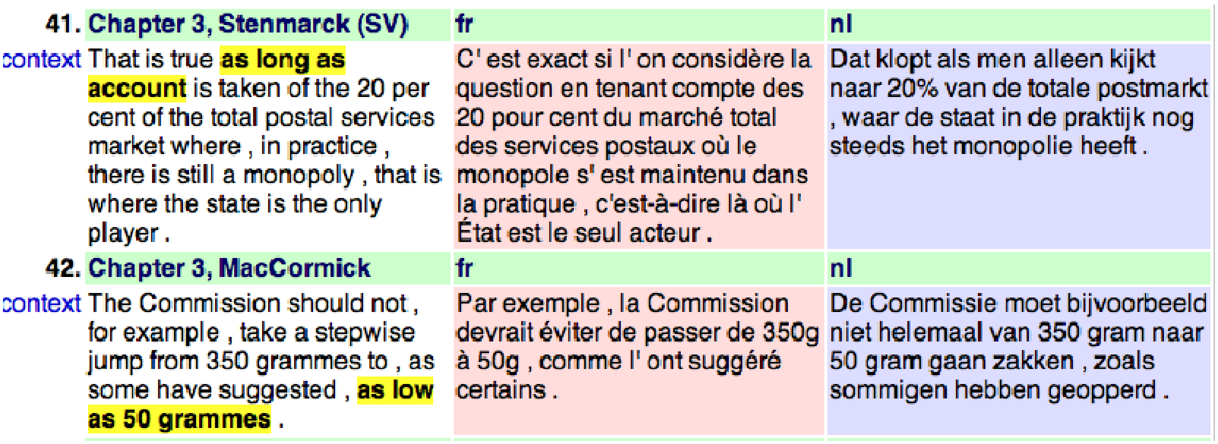
[3] Available at `http://sites.google.com/site/adwaitratnaparkhi/publications/jmx.tar.gz?attredirects=0`, developed by Adwait Ratnaparkhi

3.3.2 Europarl: European Parliament Proceedings Parallel Corpus

http://www.statmt.org/europarl/

The Europarl corpus [29] consists of parallel text proceedings of the European Parliament in 11 languages[4]: Danish, German, Greek, English, Spanish, Finnish, French, Italian, Dutch, Portuguese and Swedish. The corpus was compiled mainly to aid machine translation, but has since been used to solve many other problems in NLP such as word sense disambiguation, anaphora resolution, information extraction etc. In comparison to previously mentioned Hansards corpus, the researchers that worked on the Europarl corpus gathered data as a third party from the website of the European Parliament by using web crawlers. Once acquired, the raw text from the HTMLs is also extracted using a Perl script which was adapted to the special format of the proceedings where e.g. speakers are separated with special tags. The next step, extracting sentences, is more complex since their splittings and tokenization require special tools for each language, and they are not always available. Consequently, the existing sentence splitters and tokenizers were adapted in a semi-automatic way in order to achieve better performance. In the case of Europarl, sentence alignment was made significantly easier because the texts were already, due to well formed HTML structure, divided into paragraphs. This made it possible for the *Gale-Church* [22] algorithm to work efficiently, since it is solely based on the comparison of the character length between sentences. The number of sentences per paragraph was small, therefore the algorithm's precision was high, even managing to align one-to-two or two-to-one sentences combinations when coming across one long sentence and two short sentences and vice versa.

[4] Official languages of the European Union at the time of the publication of Europarl (2005)



41.	**Chapter 3, Stenmarck (SV)**	**fr**	**nl**
context	That is true **as long as account** is taken of the 20 per cent of the total postal services market where , in practice , there is still a monopoly , that is where the state is the only player .	C' est exact si l' on considère la question en tenant compte des 20 pour cent du marché total des services postaux où le monopole s' est maintenu dans la pratique , c'est-à-dire là où l' État est le seul acteur .	Dat klopt als men alleen kijkt naar 20% van de totale postmarkt , waar de staat in de praktijk nog steeds het monopolie heeft .
42.	**Chapter 3, MacCormick**	**fr**	**nl**
context	The Commission should not , for example , take a stepwise jump from 350 grammes to , as some have suggested , **as low as 50 grammes** .	Par exemple , la Commission devrait éviter de passer de 350g à 50g , comme l' ont suggéré certains .	De Commissie moet bijvoorbeeld niet helemaal van 350 gram naar 50 gram gaan zakken , zoals sommigen hebben geopperd .

Figure 3.1: Example of an entry in the Europarl corpus [61]

3.3.3 MultiUN: A Multilingual Corpus from United Nation Documents

http://www.dfki.de/lt/publication_show.php?id=4790

MultiUN [19] compiled of around 300 millions words per 6 languages of United Nations, namely Arabic, Chinese, Russian, Spanish, English and French. The collection of texts, as was the case with Europarl, was collected by using web crawlers. Afterwards, as the data was mostly encoded in the Microsoft Word ODS format, it needed to be cleaned up and adequately processed to be converted to XML. This consisted of the removal of certain content like images, tables and style markers. The authors point out that not all extracted text is suitable for further language processing, and while it is good to acquire as much text as possible, what is more important to ensure the quality of the texts[5]. Furthermore, the sentence extraction was achieved by the Natural Language ToolKit (NLTK) [10], while

[5] During the work I did on pairing the English and Serbian sentences from the corpus I had available, it sometimes occurred that during the step of preprocessing some of the text was lost. However, in the long run this still meant that in the significantly shorter amount of time, more aligned sentences would be obtained than if the same process was achieved manually. Additionally, the text needed to be cleaned up in a similar way of removing images, tables, reference markers etc.

the Chinese sentences were segmented by using regular expressions. Finally, the extracted sentences were automatically aligned by `hunalign`, which is discussed in more detail on pages in section 3.2.2.

3.3.4 JRC-Acquis

`https://ec.europa.eu/jrc/en/language-technologies`
The JRC-Acquis [56] corpus was developed with the motivation to extend the previously published Europarl [29] corpus, namely add the languages of new Member States, as well as candidate countries. As of January 2014 there are 24 official languages of the European Union: Bulgarian, Croatian, Czech, Danish, Dutch, English, Estonian, Finnish, French, German, Greek, Hungarian, Irish, Italian, Latvian, Lithuanian, Maltese, Polish, Portuguese, Romanian, Slovak, Slovene, Spanish and Swedish [56]. In 2017 this number remains the same[6]. In the pre-processing step, web crawlers were used to download the HTML documents which were subsequently converted to UTF-8 encoded XML format. The HTML files initially had well structured sections in terms of paragraphs and line-breaks, and the formatting was consistent across all languages so the structure could be preserved later on and facilitate the sentence segmentation. The authors do not specify the way they extracted the sentences. It can be assumed that it was done in the way similar to the Europarl [29] corpus where the sentence splitters were adapted to each language. However, it would be useful to see the methods that were used in this step because of the number of different languages that were used and which probably do not all have the same amount available tools and resources. The final step of aligning the sentence was done by both *Vanilla* [17] and `hunalign` [64]. The *Vanilla* aligner is an implementation

[6] `http://ec.europa.eu/education/official-languages-eu-0_en` Accessed 16 July 2017

of the *Gale-Church* algorithm [22], hence uses purely statistical methods. The user has an option to choose the output of either of the aligners.

3.3.5 OPUS

`http://opus.lingfil.uu.se/`

The OPUS [62] corpus is a big collection of translated documents which are downloaded from the World Wide Web. Currently, there are about 30 million words in 60 languages. The corpus is sentence aligned and aims to acquire linguistic annotation, as well. The corpus is based on open-source distributions and their documentation such as OpenOffice.org documentation, Ubuntu, GNOME, PHP manuals etc. The authors say that linguistic markup contains sentence boundaries, word boundaries, part-of-speech tags (for certain languages) and shallow syntactic structures (for English in some parts of the corpus). However, the authors do not specify the way the sentences were extracted, but I believe it is implied that on the basis of previous structure of the documentation it was possible to extract the sentences easily and keep the paragraph boundaries, as well. All files are UTF-8 encoded and are sentence aligned using the *Gale-Church* algorithm [22].

The advantage of this corpus is that it has a significantly larger number of language pairs than the other corpora, but the domain still remains very limited and seeing it exploits the open source data, the quality of the translation might be expected to be worse than the translation done by professionals.

3.3.6 MULTEXT-East

`http://nl.ijs.si/ME/V4/`

From the mid-nineties onwards, the languages of the European Union gained momentum in development of multilingual resources, especially through funding of the EU projects. On the other hand, the same level of progress was not made for the languages of Central and Eastern Europe, wherein lies the motivation for the compilation of MULTEXT-East project [20]. The project contains several different corpora, however the most valuable one turned out to be fully annotated corpus of George Orwell's novel *1984* in the English original and translations: Romanian, Slovene, Czech, Bulgarian, Estonian, Hungarian, Latvian, Lithuanian, Serbian and Russian. The sentences were automatically aligned and then manually verified, which was possible since the novel is approximately 100 000 words long. No further specifications of the methods were mentioned in the paper since each language was associated with separate contributors. In a related paper for Serbian [34] the authors mention the use of `Unitex` [50] for detecting sentence boundaries. For the alignment, it can be assumed that an implementation of a Church-Gale algorithm was used once again and then manually verified. Since the resource was a novel, we can expect the structure to be well defined in terms of chapters and paragraphs. This project differs in comparison to other corpora seen thus far because the parallel texts were a secondary result, while the primary motivation was to gather linguistically annotated data.

Table 1: Methods used in available parallel corpora

Corpus	*Language combination*	*Text extraction*	*Sentence boundary detection*	*Sentence alignment*
Hansards	English, French	HTML::Parser Perl script	MXTERMINATOR	GSA Tool
Europarl	11 official languages of the European Union (2005)	Unpublished Perl script	Ad-hoc solutions for each of the languages	Implementation of the *Gale-Church* algorithm
MultiUN	6 languages of the United Nations	ODS to XML conversion	NLTK and regular expressions	hunalign
JRC-Acquis	24 official languages of the European Union (2014)	HTML to XML conversion (UTF-8 encoded)	Not specified; possible that the same techniques as in Europarl were used	Vanilla and hunalign
OPUS	60+ languages present on the World Wide Web	Source format not specified, converted to XML	Not specified; possible that the structure of documentation was exploited	Implementation of the *Gale-Church* algorithm
MULTEXT-East project	17 languages of Central and Eastern Europe (across different corpora)	Formats not specified	Not specified for every language; `Unitex` used for Serbian	Not specified; possible implementation of the *Gale-Church* algorithm; output manually verified

3.4 Algorithms for sentence alignment

Sentence alignment algorithms can roughly be divided into three types: length-based, lexicon-based and hybrid of the two. Length-based algorithms take the length of sentences as an input, measured by the number of words or the number of characters [39]. The most used algorithm of such type is the *Gale-Church* algorithm [22], which uses the character based length measure. These algorithms are fast and have an overall good performance if there is minimal noise. Lexical-based algorithms rely on lexical constraints to perform the alignment. They use previously made available lexical information of source and target language to determine the relationship between the source text and its translation [40] [16]. Even though this approach is more robust and deals with noise more efficiently, it is much more computationally expensive and requires extensive resources. These two approaches can be combined to reach optimal results in two ways. The first one is to use both length and lexicon information at the same time and determine the alignment [54]. The second one uses the two approaches subsequently. Firstly, the length-based algorithm is applied and its result is filtered and verified by the lexicon-based algorithm to achieve a more precise output [47] [64]. However, both of these approaches suffer from high computational expenses and are problematic for processing large corpora [39].

3.4.1 Cognates as anchor points

In linguistics, cognates are words which share etymological origin. More precisely, they are pairs of words in different languages which share phonological and orthographic properties, as well as semantic properties, therefore they are likely to constitute mutual translations [54]. Some examples of French and English cognates are *erreur/error*, *dictionnaire/dic-*

tionary, *distance/distance* etc. In terms of sentence alignment, the assumption is that a significant number of cognates can be found between sentences which are mutual translations. The easiest way to incorporate these cognates as anchor points is to modify length-based methods for alignment and use the number of cognates between sentences instead of the length metric. However, the algorithm works more efficiently when combined with the length information [54]. This remains an interesting approach which is not as computationally expensive as when adding lexical resources. However, it should be noted that its scope remains limited because it can be used on pairs of languages which belong to the same language family or have strong etymological connections. Even if the large influx of anglicisms is considered, it is questionable if they would always be able to be captured due to different orthographic variations. For example if we consider some English and Serbian cognates like *computer/kompjuter, weekend/vikend, container/kontejner*, we can see that the difference in writing is significant. On the other hand if resources are available in languages which are historically close, it would make sense to apply this method. Finally, when working with cognates, one should also be aware of *faux amis* which are words that superficially look related, but have different semantic properties. Some examples of French and English faux amis are *blessé/blessed, envie/envy, coin/coin, passer/pass* etc. It is true that etymologically these words are most likely connected, but language change led to the different semantic properties of these words. Consequently, these words may indicate inaccurate alignments. This is an important caveat to consider, especially because faux amis are often words which are high in frequency.

3.4.2 Neural networks for sentence alignment

Following the general trend in NLP, a number of methods for aligning sentences using neural networks were proposed. The approach was especially popularized by using word embeddings which are able to capture the meaning of words in vector space. The sentences can be aligned by comparing the sequences of these representations and matching them. Even though the learning of these vector representations is computationally intensive, they can be reused over time which makes the process more acceptable. However, the neural networks for learning these embeddings require a lot of data to be trained on, which are not always available.

3.5 Examples of alignment methods

3.5.1 Length based: Gale-Church

An algorithm representative of the length-based sentence aligning is the *Gale-Church* [22] algorithm. It is a classic algorithm mentioned in literature for unsupervised sentence alignment. It is language independent and considering its simplicity, it is "remarkable that such a simple approach works as well as it does" [22], therefore we encounter its implementation very often. The idea behind the algorithm is that every bitext will have the same characteristic: matching segments will be of similar length, i.e. "longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter sentences [22]." The *Gale-Church* algorithm uses dynamic programming to resolve the alignment. All the information the algorithm receives as an input is the length (number of characters) of source sentences and target sentences. In the beginning the matrix of all partial solutions is created and is filled from top to bottom diagonally. The *edit distance* algorithm is actually analogous to the *Gale-Church*, where the *edit distance* compares strings and the *Gale-Church* compares sequences of strings (sentences) by using three operations of *insertion*, *deletion* and *substitution*. Moreover, the *Gale-Church* algorithm allows two additional operations: *one-to-two alignment* and *two-to-one alignment* which makes it possible to connect single sentences that were translated into two sentences and vice versa. Each of the sentence pairs is assigned a probability score based on those measures and the most likely outcome is selected by using the *maximum likelihood* measure. The creators of the algorithm performed tests on English-French and English-German data with an error rate of only 4.2% on 1316 alignments. An interesting thing to note is that the authors chose to go for the character-based length instead of word-based length, even though intuitively it would seem more appropriate to use the number of words in a sentence to examine the similarity. However, the authors claim that there is less variability in the number of characters than in the number of words. On the other hand, if we take a look at a language that is prone to long compounds, like German or Dutch, it does make more sense to take the character number into consideration, because a word like *Haustürschlüsselloch* will be translated into three words with *house door keyhole*, but will remain similar with the character length of 20 and 18 (*Gale-Church* includes whitespaces in the number of characters).

Does this rule adequately apply to English and Serbian sentences? Table 2 represents some randomly chosen examples from the aforementioned MULTEXT East Project *1984* corpus [34]. The table contains the example sentences, number of words and characters for each sentence and the result of the *Gale-Church* metric σ.

Table 2: English and Serbian sentences from George Orwell's *1984*

Example sentences	*# of words*	*σ*	*# of chars*	*σ*
E: But it had also been suggested by the book that he had just taken out of the drawer.	18	0.277	85	**0.135**
S: No na to ga je bila podstakla i sveska koju je upravo izvadio iz fioke.	15		71	
E: It was a peculiarly beautiful book.	6	**0.129**	35	0.142
S: Bila je neobično lepa.	4		22	
E: Its smooth creamy paper, a little yellowed by age, was of a kind that had not been manufactured for at least forty years past.	24	0.320	126	**0.102**
S: Njen gladak beli papir, nešto požuteo od vremena, bio je od one vrste koja se nije proizvodila najmanje četrdeset godina.	20		121	
E: He could guess, however, that the book was much older than that.	12	0.181	65	**0.075**
S: Međutim, nije mu bilo teško pogoditi da je sveska još starija.	11		62	
E: He had seen it lying in the window of a frowsy little junk-shop in a slummy quarter of the town (just what quarter he did not now remember) and had been stricken immediately by an overwhelming desire to possess it.	40	0.496	214	**0.223**
S: Bio ju je spazio u izlogu zapuštene male starinarnice u jednoj od siromašnih četvrti grada (nije se tačno sećao kojoj) i smesta ga je zahvatila neodoljiva želja da je poseduje.	30		176	

The results from the table indicate that the proportion of differences slightly favors the character to the word metric. Consequently, it can be expected that the *Gale-Church* algorithm will perform as good as on other languages it has been tested on.

Moreover, this can be tested out on this small corpus by a *Gale-Church* formula for calculating the σ metric:

$$\sigma = \frac{(L2 - L1c)}{\sqrt{L1s^2}}$$

- *L1* - number of characters in the source sentence
- *L2* - number of characters in the target sentence
- c - the mean, i.e. the expected number of characters in *L2* per character in *L1*; calculated by the sum of all characters in *L2* divided by the sum of all characters in *L2*
- s^2 - the variance of the number of characters in *L1* per character in *L2*; calculated by the squared sum of differences between the number of characters and divided by the total number of sentences

The application of the formula showed that in most cases the character length prevailed (lower σ result indicates higher certainty). The second sentence showed higher certainty when considering word length, but it can be expected that short sentences will have the same or very similar number of words.

In the previous section we have seen that this algorithm is applied very often in alignment task. The authors suggest that its performance could be augmented by adding some lexical constraints. In conclusion, this algorithm remains convenient because of its simplicity and the fact that it can be applied to any language, which makes a great advantage for alignment of sentences in low-resource languages.

3.5.2 Lexicon-based:Champollion

The `Champollion` [40] algorithm takes advantage of the *tf-idf* value, a metric frequently used in information retrieval. The *tf-idf* seeks to filter out terms with the highest weight, i.e. importance in the processed document. The highest weight is assigned to terms which occur many times in a small number of documents. Lower relevance is signaled if a term appears a few times in a few documents or occurs many times in many documents. The lowest importance is assigned to terms which appear in all of the documents [42]. This way the function words like articles and prepositions are taken out of the equation and keywords are given high weights. In this case a term corresponds to a word in a sentence, and a document to a sentence. Based on the weights, a score between the two sentences is calculated and the highest score results in the alignment of two sentences. The advantage of this algorithm is that it assumes a noisy input, meaning that it presupposes that a large number of sentences will not be mapped one to one, and that the number of deletions and insertions will be significant [40]. The path to the aligned sentences is done via dynamic programming algorithm, similar to the one used by the *Gale-Church*, but instead of searching for the minimum distance, `Champollion` searches for the highest similarity. This approached resulted in high precision and recall scores and produced high quality alignments. However, it has the time complexity of $O(n^2)$ and since it uses the methods of dynamic programming, it needs to look up the words in the dictionary multiple times, which makes it very slow. Additionally, there is a layer of tokenization and light stemming for both texts whose processing time may vary depending on the language.

3.5.3 Hybrid: Fast-Champollion

This algorithm is the successor to the Champollion algorithm and it aims to keep the high quality alignments, while reducing the running time. The idea behind this aligner is to split the text into smaller chunks and in that way minimize the running time while maintaining the quality of the Champollion. Smaller parts of text are obtained via length-based method, which is not robust enough, but achieves faster performance [39]. The information kept from the length-based alignment are anchor points which are considered to be reliably connected and allow the faster performance of the lexicon-based algorithm which is entirely based on the Champollion algorithm from the previous section. The results of the Fast-Champollion showed that when shorter texts were processed, it ran faster than the Champollion algorithm while obtaining higher precision and recall results and maintaining the robustness level. The Champollion, however, showed superior precision and recall results on the longer texts. Both Champollion and Fast-Champollion require large dictionary they can exploit. The authors point out that the precision and recall results drop significantly as the dictionary reduces. Therefore, these aligners can only produce high quality results for languages with extensive resources. Nonetheless, this can be expected from approaches which rely on exploiting lexical data in their alignments. One step further in this direction are approaches towards automatic creation of reliable bilingual dictionaries which may later be reused by these aligners.

3.5.4 Hybrid: hunalign

The hunalign sentence aligner is another type of hybrid-based approach. The authors of this aligner, however, kept in mind that the choice of method alignments greatly depend on what they call *density* of the languages, which is the availability of digitally stored material [64]. They point out that most available tools are adapted for high (lexicon-based) or low (length-based) density languages, whereas medium density languages fall short in both categories. There are about five hundred medium density languages, which roughly have from 2 to 50 million speakers [24]. The aligner was initially made with a motivation for creating the parallel corpus of Hungarian and English, hence its name. However, its methods are completely applicable to any pair of languages. The authors gathered data of different registers and sources (the Web, the Bible, movie subtitles, open-source documentation, reports etc.) either by web crawlers or manually and adequately converted from different formats. The outcome were two bodies of text which supposedly contained the same content in source and target language. Even though the compilation of corpora greatly relied on manual work, the authors wanted to achieve the alignment step completely automatically. The authors considered using cognates to aid the alignment, but as Hungarian and English are from different language families, the results were not adequate. That is why they opted for the use of a simple dictionary and length-based methods. In the first step the algorithm produces a crude translation of the source text based on the available dictionary. If failing to find an entry in the dictionary, the algorithm copies the word from the source text. Afterwards, the produced translation is compared against the source language sentences and the similarity score is based on the word matching and sentence length. The score for every sentence pair is calculated around the diagonal of the alignment matrix. The assumption is that the highest accuracy of alignment in unprocessed texts will be at the beginning and at the end of the document. Once the similarity matrix is obtained, the algorithm looks for the optimal path by the methods of dynamic programming. To achieve faster performance, the algorithm takes into account only 1:1, 1:2 and 2:1 sentence matches. After the path has been found, the postprocessing step iteratively merges a neighboring pair of one-to-many and zero-to-one sentences based on the length information. Using this method, any one to many mappings can be discovered. The hunalign algorithm is able to produce meaningful alignments without the presence of the dictionary, as well. If the dictionary is absent, an ad-hoc dictionary is created after the first alignment. From this step, the algorithm creates a dictionary based on the co-occurrences in the sentences. Based on an association measure between the two, a certain threshold determines which words will constitute a dictionary. The authors compare their algorithm against Moore′s alignment algorithm [47] which uses similar methods as hunalign. However, based on precision and recall metrics, the hunalign method outperforms the Moore′s. In conclusion, the hunalign is a very reliable method for creating corpora of medium density languages and shows excellent results when dealing with carefully selected, often manually preprocessed data. As seen in the previous section, since its publication, hunalign has been used to aid the compilation of many multilingual corpora because it represents a good compromise between computationally intensive lexicon-based methods and length-based algorithms which lack robustness.

3.5.5 Hybrid: Gargantua

Gargantua [11] is a two-step clustering approach for sentence alignment with emphasis on minimizing the computational cost. This is done by splitting the search for the optimal path into two parts. The idea is the sentences would form clusters and the alignment will be achieved at this level, therefore minimizing the computational cost of aligning per sentence. Overall, the two-pass unsupervised approach is done in the following way: in the beginning an approximate alignment is performed by using length-based methods. In the next step, each of found alignment is cached and there is a search for the optimal alignment considering only links made of at most one sentence in each language [72]. In the following step, the alignment is heuristically improved by using adjacent links, as well. This algorithm is actually very similar to the famous `Moore's` [47] sentence alignment, with the difference of paying additional attention to lowering the computational cost. As [72] observe, the key observation is that the searching for optimal alignment can be done very fast, but like most work of this vein, it is prone to missing large portions of untranslated text.

3.5.6 Semantic embeddings for sentence alignment

Semantic embeddings are vectors whose relative similarities correlate with semantic similarity and are based on the idea that contextual information alone is a viable representation of linguistic items [2]. This task is closely related to statistical language modeling, but while the traditional techniques used hand-crafted features to create word embeddings, the vectors inferred by artificial neural networks [9], do not require this manual work. In the beginning, the feedforward neural networks were used and were able to learn an appropriate set of features, while predicting the next word in a sentence, all the while carrying important linguistic information. This has raised an interest in using neural networks for further language modeling [18] [9]. One of the main advantages of semantic embeddings is that they are able to process a vast of readily available data that became more accessible with the growth of the Internet, social networks and media records, along with huge increase in computational resources. This offered a solution to a previous problem of statistical language modeling which required a lot manually annotated data and feature engineering.

However, due to the nature of the way these embeddings are obtained, they neglect word order and the principle of linguistic compositionality and focus on the Bag-Of-Words approaches. With this in mind, Le & Mikolov (2014) [45] [35] propose the *Paragraph Vector*, an unsupervised algorithm that learns fixed-length feature representations from pieces of text with various length (sentences, paragraphs, documents). In other words, by using this method, it is possible to obtain sentence embeddings where the meaning of each sentences will be represented as a numerical value in the vector space. Once all the sentence embeddings are learned, their values can be compared and the pairs of sentences can be extracted.

Even though this approach of obtaining sentence alignments is different from the standard methods found in literature, there are certain advantages to it which should not be neglected. Firstly, the embeddings capture meaning in the numerical form which suggests it is completely language independent. Secondly, the embeddings would be able to learn much more than simple matching, they would be able to capture deeper linguistic relations like synonymy, antonymy, paraphrasing etc. From the point of view of a parallel corpus of sentence embeddings, they would play a significant role in cross-language information retrieval and machine translation, especially in the domain of low-resource languages where the collection and the creation of necessary data would take an extensive amount of time [38]. The research in this area is still in its early days, therefore the approaches and the terminology are still scattered which may cause theoretical inconsistencies. Nevertheless, we can expect that semantic embeddings will only flourish with more data made available.

Chapter 4

EXALP: Extraction and Alignment Pipeline

After I had finished my internship I had a clear overview of all the steps that need to be taken in the process of compiling parallel corpora. As seen in Chapter 2, a lot of the work was done manually and it was evident that reaching a corpus of satisfactory size would take a significant amount of work and time. Moreover, each step makes part of a pipeline where every process is dependent on the previous one and involves several people. After doing some research I had an idea of making a program which would only take two files as an input and would produce an aligned text file as an output. I decided to work with `python 2.7` because it has a lot of NLP related open-source resources available and a large online community which helped me cut the learning curve, as I had not worked with `python` previously. In the beginning, I was a bit discouraged about the first step of text extraction from different formats (mostly `.pdf`) because various structures of potential files to be processed would make the program less robust. However, I decided to proceed with that step as well, and after testing out a few methods, I found the one whose outcome was satisfactory. The raw text is cleaned up and adapted for later use, which involves some regular expressions for replacing certain characters or deleting superfluous data. Once the text is cleaned up, the segmenting phase ensues. At first I wanted to use the NLTK [10] package, the most famous platform for working with natural language data with python. However, after I tested it out on a set of English sentences, it turned out that its sentence tokenizer was not as reliable as I had hoped[1]. NLTK has the option of training the tokenizer for a language other than English, however, since the English sentences were not accurately tokenized and I did not have enough data for the training of Serbian sentences, I decided to work with Unitex. In fact, Unitex is the only open-source platform which has an active and updated support for Serbian. The graphs Unitex uses to segment the sentences for both English and Serbian gave plausible results. A different challenge was incorporating the graphs to work with python what I managed with a package which provides access to Unitex's C++ library on github[2]. This is the first step that is actually language dependent - graphs must correspond to the language being processed. In the next step, I adapted the Perl script I had made during my internship which creates an XML file and incorporated it. Final preprocessing step was to have a file which has one sentence per line, because that is the format of the input most aligners require. The next step is the alignment of sentences. For the reasons mentioned in its section in Chapter 3, I chose to work with hunalign and incorporated as a final step. Alternatively, I also included Gargantua aligner for the sake of comparison between methods. Additionally, any new sentence alignment method can be added as an alternative. Taking into consideration time constraints and facility of implementation,

[1] For example, the tokenizer split the sentences after certain abbreviations like *e.g.*

[2] https://github.com/patwat/python-unitex Accessed July 28 2017

these were the two methods I worked with. For the sake of readability, I incorporated an `.html` file which transforms the output into two-column table and accompanying `.css` file.

For the reason of illustrating the process, let's say I wanted to align two documents `abc_en.pdf` in English and `abc_sr.pdf` in Serbian. In fact, the files ought to have a distinctive language marker before the extension (in this case `en` and `sr`). The program is ran with the following command: `python EXALP.py abc_en.pdf abc_sr.pdf` The output files are then sorted as shown in Fig. 4.1.

4.1 Steps of **EXALP**

4.1.1 Input arguments

`EXALP` takes two files as an input. At the moment, the text files can be in the `.pdf` or the already raw `.txt` format. If the file is a `.pdf`, it will proceed to the text extraction step. Otherwise, it will skip it.

It is important for the names of the files to be consistent. The obligatory format is a language marker before the extension (`en` for English, `sr` for Serbian; currently it is also possible to have `de` for German and `fr` for French, although these language pairs are still untested). The rest of the filename is arbitrary, but consistency[3] is recommended because the folder tree structure will be based on the filenames.

[3] preferably the files have the same name except for the language marker

```
abc
|
+- abc_en
|  |
|  +- abc_en.xml
|  |
|  +- abc_en_extracted.txt
|  |
|  +- abc_en_snt.txt
|  |
|  +- abc_en_unitex.txt
|
+- abc_sr
|  |
|  +- abc_sr.xml
|  |
|  +- abc_sr_extracted.txt
|  |
|  +- abc_sr_snt.txt
|  |
|  +- abc_sr_unitex.txt
|
+- abc_dictionary.dic
|
+- abc_en_abc_sr_hun.html
|
+- abc_en_abc_sr_hun.txt
|
+- abc_en_abc_sr_garg.html
|
+- abc_en_abc_sr_garg.txt
|
+- lrstyle.css
```

Figure 4.1: `EXALP`'s output as a structure of folders

4.1.2 Raw text extraction

If the input file is in the `.pdf` format, it is necessary to extract the text from it in order to process it in the further steps. This is not a trivial task because of the nature of the `.pdf` file which encapsulates a complete descripton of a fixed-layout document, including the text, fonts, graphics and other information needed to display it[4]. To deal with this step I worked with `python`'s `textract`[5] library which is able to extract content from any type of file, without any irrelevant markup. I chose this particular library because it supports various formats which makes it convienient for adding more types of formats in the future. The output of the processed `.pdfs` is reliable, the only downside in comparison to the manually extracted text is that the structure of paragraphs is lost. However, as I was mainly concerned with aligning the sentences, this did not pose any problems in the steps that follow.

4.1.3 Cleaning up the raw text

In this step the `EXALP` cleans up the processed text in the following way:

- Hyphenation

  It removes the hyphenation in broken lines and connects:

  ```
  A centralized manage-
  ment of the networks can be conducted [...].
  ```

  becomes

  ```
  A centralized management of the networks can be conducted [...].
  ```

[4] `https://en.wikipedia.org/wiki/Portable_Document_Format` Accessed 29 July 2017

[5] `https://textract.readthedocs.io/en/stable/` Accessed 29 July 2017

- Deleting page numbers
- Removing empty lines
- Correcting letters with diacritical marks

  Serbian has the following letters with diacritical marks: č, ć, đ, š, ž. They are not correctly converted with `textract`, but they are always converted in the same way (e.g. Ê is Š, | is đ), so I used a set of regular expressions and replaced these characters with correct letters.

The output is the `*_extracted.txt` which contains processed raw text from the input file.

4.1.4 Processing with `Unitex`

I used `python` bindings for `Unitex/GramLab` I found on `github`[6] which allowed me to use `Unitex` as a library and call `Unitex` functions from `python`. I compiled the graphs `Unitex` uses for sentence segmentation for English and Serbian. This is the first step which is language dependent (if we exclude the correction of letters with diacritical marks, which is applicable to all languages that use them). The advantage of using these graphs for sentence segmentation is that they can be easily modified if necessary by adding more special cases. In the work I did thus far no modification of these graphs was necessary, as it segmented the sentences correctly. Additionally, `Unitex` has support for Arabic, Finnish, French, Georgian, German, Greek, Italian, Korean, Latin, Malagasy, Norwegian, Polish, Portuguese, Russian, Spanish and Thai. This means that the graphs for any pair of these languages can easily be added by writing an additional line of code in `EXALP`.

[6] `https://github.com/patwat/python-unitex` Accessed 29 July 2017

Currently I added an option for French and German to use with adequate fr and de language markers.

The output is the *_unitex.txt file which contains a {S} delimiter for every sentence.

4.1.5 Creating an XML

For this part I used a script I made in Perl during my internship and adapted it to work with the rest of EXALP. This step creates an XML where {S} delimiters are used to segment each sentence between two <seg></seg> tags.

The output is the *.xml file.

4.1.6 Sentence per line

This step uses the XML file to create a file which has one sentence per line. This is necessary because most aligners require this kind of input.

The output is the *_snt.txt file.

4.1.7 hunalign

In this step hunalign is called and two *_snt.txt files are an input (one file per language). hunalign offers different options which are chosen via arguments given in the command line. The output can be either in the numerical format or in the text format. In the text format, each line is separated by tab into three columns. The first column corresponds to source text sentences. The second column contains corresponding target text sentences. The third column is hunalign's internal metric for confidence level. In the case of merging sentences which correspond to one sentence, the merging point is indicated by the separating sequence ~~~. In the numerical or *ladder* format output, the output file does not contain the sentences, but their indices. It also contains three columns, where the third one is the confidence level. The format used by EXALP is exclusively the text format output.

hunalign's dictionary argument

One of hunalign's obligatory arguments is the dictionary which contains dictionary entries separated by a newline. However, without the presence of a dictionary, hunalign can still produce the alignment with the null.dic argument. In that case the alignment is based on the character length and each entry has the word in the source language, followed by a whitespace, '@', another whitespace and the word in the target language. The dictionary may contain multiword units, as well. For example, the English-French dictionary has the following format:

```
1st @ 1er
a cappella @ a cappella
a drop in the bucket @ une goutte dans l'océan
a little @ un peu
a lot @ beaucoup
a lot @ souvent
a priori @ a priori
```

The dictionary files did not come with the hunalign package, it only contained a sample of English-Hungarian dictionary. However, I found other sample dictionaries which are used by the `LFAligner`[7], an open-source software aimed at professional translators for creating translation memories. `LFAligner` uses `hunalign` in the background and has a number of dictionaires which I found in its source files. At this point I was still at an impasse because there was no combination which contained Serbian. I extracted the column of English entries and started translating it manually, but as the dictionary had around 20 000 entries I looked for a way to do it more efficiently. I decided to use the `Google translate` option of translating the whole document[8]. I took the output file and combined it with the English file in the desired dictionary format. This dictionary has its faults, mostly because it is unable to capture all the flective forms of the word. Ideally, a dictionary should have the following entries:

```
apple @ jabuka
apple @ jabuke
apple @ jabuci
apple @ jabuku
apple @ jabuko
apple @ jabukom
```

whereas now it only contains:

```
apple @ jabuka
```

This is a case not only for nouns, but for verbs as well. Another issue is that the English dictionary provides multiple entries for the same word or multiword unit. The example from the English-French dictionary:

```
face @ visage
face @ face
```

[7] `https://sourceforge.net/projects/aligner/` Accessed July 30 2017

[8] `https://translate.google.com/` Accessed July 30 2017

```
face @ figure
face @ affronter
face @ faire face à
(...)
```

whereas in the English-Serbian version, because of the way the statistical machine translation works, every repeating entry is always assigned the same translation:

```
face @ lice
face @ lice
face @ lice
(...)
```

Towards the end of the redaction of this thesis, my internship supervisor provided me with an English-Serbian dictionary DELAF, which also contains the flective forms. Compiled this way, the DELAF dictionary has over 100 000 entries. For the results and impact of the dictionaries, see section 4.2.3.

`hunalign`'s realignment

One of the `hunalign`'s options I used was the `realign` option. If this option is activated, the alignment is built in three phases. After the initial alignment, based on the character length, the algorithm adds dictionary entries based on the co-occurrences in the identified aligned sentences. Afterwards, the alignment step is repeated based on this expanded dictionary. That dictionary is saved as `*.dic` file.

4.1.8 Gargantua

In the same way hunalign is called within EXALP, the next step is to do the same with the Gargantua method. Gargantua takes the same input as hunalign (one sentence per line). The output, however, slightly differs from the hunalign's output. The only possible output file is the one that corresponds to hunalign's *ladder* format, meaning that the output file only contains indices of paired sentences. I wanted to have the same type of output for both aligners, so I would be able to compare them. To deal with this, I created an additional function that creates text files with sentences which correspond to indices obtained with Gargantua. Overall, the implementation of Gargantua posed a lot of problems in comparison to hunalign because there was less documentation available.

4.1.9 Conversion to .html

In this final step the output files from both hunalign and Gargantua are made more user-friendly by conversion to an .html format with corresponding .css file. Considering hunalign's confidence levels, the table is either white and blue for high confidence level, brown for medium confidence level and red for low confidence level. Gargantua did not have this metric included, so its output is only in white and blue.

4.2 EXALP: Evaluation

In this phase the EXALP_evaluation.py was made to evaluate the results obtained by EXALP using the measures of precision, recall and F-score.

Recall is a measure of how much relevant information the system has extracted from the text [28]. In the case of EXALP, recall is defined in the following way:

$$Recall = \frac{\#\ of\ extracted\ sentences}{total\ \#\ number\ of\ sentences\ to\ be\ extracted}$$

Precision is a measure of how much of the information that the system returned is actually correct. This measure is also known as accuracy [28]. In the case of EXALP, precision is defined in the following way:

$$Precision = \frac{\#\ of\ correctly\ aligned\ sentences}{total\ \#\ of\ \ correctly\ aligned\ sentences}$$

F-score is a weighted average of precision and recall, where 1 is the best value and 0 the worst. The traditional F-score is the harmonic mean of precision and recall:

$$F - score = 2 \times \frac{1}{\frac{1}{recall} + \frac{1}{precision}} = 2 \times \frac{precision \times recall}{precision + recall}$$

The evaluation of EXALP was calculated against the manually extracted and aligned data gathered throughout the summer of 2016. The data consists of aligned sentences of the *Management*[9] scientific journal.

[9] http://management.fon.bg.ac.rs/index.php/mng Accessed 28 July 2017

The evaluation is done on three test datasets, presented in Table 3:

Table 3: Datasets for evaluation

Dataset	*Management Journals*	*Number of journals*	*Number of alignments*
A	57-58	2	2922
B	51-54	4	5831
C	47-61	15	21286
D*	61-64	4	7880

4.2.1 The evaluation process

In the first phase, the recall is measured. The goal is to count the sentences that occur in automatically extracted data that match the sentences from test datasets. This is not trivial because it is not possible to use simple string matching. The issues occurred because some of the data that was manually extracted still contained hypehnation dashes, the diacritical letters were manually converted and were sometimes written as *c* instead of *č* and *s* instead of *š*. These occurrences are successfully dealt with in the `EXALP` phase of correcting the raw data. Furthermore, any additional whitespaces or punctuation marks would lead to a mismatch, while in reality the sentence should be regarded as correctly extracted data.

Because the results with simple string matching gave unrealistically low recall result, an approximate, instead of total, string matching was made. This was achieved with the `fuzzywuzzy`[10] module for fuzzy string matching which relies on *Levenshtein distance* to

[10] urlhttps://github.com/seatgeek/fuzzywuzzy Accessed 5 August 2017

match strings.

Levenshtein distance

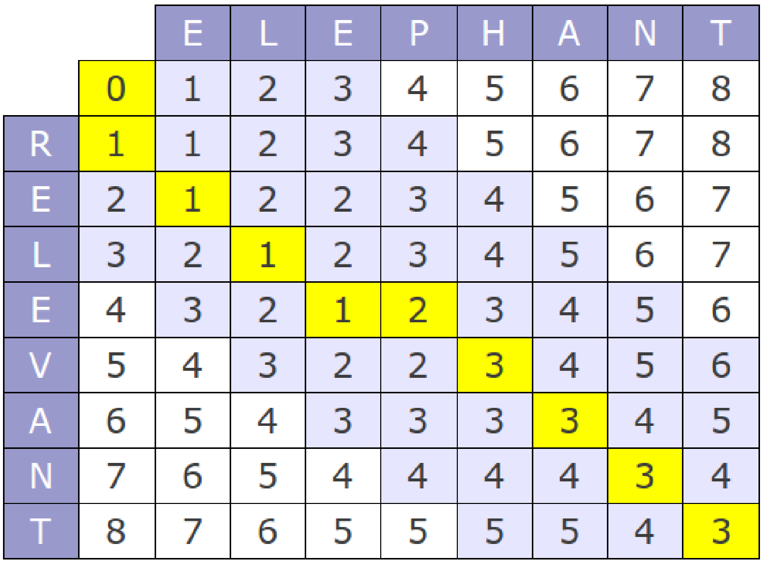
Levenshtein distance is a measure of similarity between two strings which is calculated through number of deletions, insertions and substitutions required to convert source string to target string [11] (Fig. 4.2). The distance is counted by the number of these operations performed to achieve an equal match. Greater Levenshtein distance indicates a bigger difference between the strings. The time complexity of the algorithm is $O(|s1| \times |s2|)$, i.e. $O(n^2)$ if the strings are approximately the same length[12]. Consequently, calculating the distance between a large number of strings made the evaluation process computationally intensive.

Table 4 shows some examples that were matched correctly using the `fuzzywuzzy` module. The first row shows an example where the difference is between **c** and **č**, the second one resolves the hyphenation and quotes issue and the third one deals with punctuation. The tolerance between two strings was set to 80, which means that the strings with the Levenshtein distance between 0 and 20 are acceptable matches.

[11] `https://people.cs.pitt.edu/~kirk/cs1501/Pruhs/Spring2006/assignments/editdistance/LevenshteinDistance.html` Accessed 6 August 2017

[12] `http://users.monash.edu/~lloyd/tildeAlgDS/Dynamic/Edit/` Accessed 6 August 2017

[13] Taken from `http://blog.vjeux.com/2011/c/c-fuzzy-search-with-trie.html` Accessed 6 August 2017



		E	L	E	P	H	A	N	T
	0	1	2	3	4	5	6	7	8
R	1	1	2	3	4	5	6	7	8
E	2	1	2	2	3	4	5	6	7
L	3	2	1	2	3	4	5	6	7
E	4	3	2	1	2	3	4	5	6
V	5	4	3	2	2	3	4	5	6
A	6	5	4	3	3	3	3	4	5
N	7	6	5	4	4	4	4	3	4
T	8	7	6	5	5	5	5	4	3

Figure 4.2: Example of a Levenshtein distance matrix between the words RELEVANT and ELEPHANT[13]

Table 4: Examples of strings which were fuzzy matched

S from manually aligned data	*S from automatically aligned data*	*LD*
Kupac računara, na primer, **c**esto nije svestan da je proizvođač prethodno ostvario najbolju kombinaciju delova za taj ra**c**unar u globalnom lancu snabdevanja koji povezuje veliki broj zemalja.	Kupac računara, na primer, **č**esto nije svestan da je proizvođač prethodno ostvario najbolju kombinaciju delova za taj ra**č**unar u globalnom lancu snabdevanja koji povezuje veliki broj zemalja.	4
The rules of “adequate“ and “inadequate“ behaviour should be established and clear to the parties and be approved of by the parties, i.e., the negotiators agree to play by those rules.	The rules of adequate and inadequate behaviour should be established and clear to the parties and be approved of by the parties, i.e., the negotiatiors agree to play by those rules.	13
DODATAK (B): Rezime mera)	DODATAK (B): Rezime mera 1.)	8

The precision measure verifies if the correctly extracted sentences are also aligned with their respective pair. This is also achieved by using the fuzzywuzzy module which now compares pairs of sentences.

Finally, both measures are used in the F-score formula to calculate the weighted average.

4.2.2 Results

Table 5 represents evaluation results of EXALP using the hunalign's algorithm. In these testings, hunalign used the Google Translate dictionary and the realign option (see section 4.1.7) .

Table 6 represents evaluation results of EXALP using the Gargantua algorithm. Dataset C is not included because Gargantua was not able to process large number of sentences.

Table 5: EXALP results - hunalign

Dataset	*Precision*	*Recall*	*F-score*
A (small)	0.94	0.79	0.86
B (medium)	0.91	0.72	0.81
C (large)	0.94	0.78	0.85
D* (medium)	0.94	0.78	0.85

In hunalign's results, I noticed that the medium set B had lower results than the smaller and larger dataset, therefore I included another medium set D to see if the result

[13] Taken from http://blog.vjeux.com/2011/c/c-fuzzy-search-with-trie.html Accessed 6 August 2017

was low due to the size of the dataset or its content. Considering that the dataset D's results are in the same frame as A and C, it strengthens the point that lower results were caused by the content. Otherwise, hunalign shows consistent results over all datasets.

Table 6: EXALP results - Gargantua

Dataset	*Precision*	*Recall*	*F-score*
A (small)	0.92	0.78	0.84
B (medium)	0.90	0.70	0.79
C (large)	/	/	/

Overall, hunalign showed better results than Gargantua. Additionally, hunalign was able to process the large dataset of 21286 sentence pairs, while Gargantua crashed in the process. It should be noted that the largest dataset of 28482 sentence pairs (not included in Table 5) which contained all of the manually extracted and aligned data was not able to be processed by either hunalign or Gargantua.

4.2.3 Dictionary variations with **hunalign**

This subsection considers the results obtained by varying the dictionary option in hunalign. Dictionaries that were used were the Google Translate dictionary and the augmented DELAF dictionary. Additionally, hunalign's performance was tested without using the dictionary (NULL). Another criterion was the *realign* option, therefore all the dictionaries were tested with and without the realignment. The dictionaries were tested using the dataset B.

Table 7: EXALP results - hunalign dictionary variations

Dictionary	*Realign*	*Precision*	*Recall*	*F-score*
DELAF	False	0.72	0.92	0.81
DELAF	True	0.72	0.92	0.81
Google Translate	False	0.71	0.91	0.80
Google Translate	True	0.72	0.91	0.80
NULL	False	0.72	0.86	0.78
NULL	True	0.72	0.91	0.80

The results show that the choice of the dictionary does not change the results significantly, and that it is the *realign* option that has the most influence. This additionally relieves the issue of low-resources. For reminder, the *realign* option creates a dictionary automatically after the first alignment which is based solely on the length of the sentences. This dictionary is based on the co-occurrences of the words across data.

4.2.4 Improving the results

The results imply that recall has room for improvement, as most state-of-the art results show the recall range from 0.8 to 0.9 [65], whereas EXALP's recall is at 0.79. However, the recall does not depend on aligner as much as it does on the extraction step. Therefore, improving the recall would require more work on the raw text extraction, eliminating noise, as well as more detailed analysis of sentence delimiters. Better recall results would have direct impact on improving precision scores, as well.

4.2.5 Run time

The tests were performed with `Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz`

Table 8: `EXALP` run time

Dataset	*Extraction*	*hunalign*	*Gargantua*	*Total*
A	2.71 s	0.31 s	12.16 s	15.16 s
B	91.26 s	11.13 s	3012.61 s	3115 s
C	370.38 s	114.82 s	/	485.20 s

Note that `Gargantua` is not able to process the largest dataset, therefore it is excluded from the last test.

Chapter 5

Applications of extracted bilingual data

This chapter will briefly summarize the possible applications of the output produced by EXALP. The possible use are various, from compiling bilingual dictionaries, analyzing and aiding human translation, enhancing translation memories, motivating language learners and providing interesting insignts in the domain of contrastive linguistics.

5.1 Lexicography and terminology

5.1.1 Bilingual dictionary compilation

Exploiting bilingual parallel data can automatize and accelerate the dictionary building process, which is particularly helpful in the cases of medium-density language pairs where the investment in the production of such dictionaries is often deemed unprofitable [59]. Using existing monolingual dictionaries, wordnets or monolingual corpora can facilitate the initial stage of the extraction of relevant lexical units. Parallel bilingual data can be used in the next step of finding a relevant translation for each lexical unit.

The experiment described in [59] showed that the inter-annotator agreement is surprisingly low in the domain of lexical units sense-tagging. Moreover, several other analyses implied the same outcome, which clearly shows that human annotated databases of meanings are not always reliable for finding relevant meanings. An important advantage of automatizing the process through parallel bilingual data, is removing the human intuition and building dictionaries that are completely corpus-driven and rely solely on data. This emphasizes the importance of representativeness in relation to frequency of the lexical units [57], which subsequently leads to higher quality dictionaries which stray away from prescriptivism.

5.1.2 Discovering morphological and semantic relations between lexical units

In the process of building a bilingual dictionary by making alignments between words and phrases, various links between words are created [61]. It is assumed that these links may imply morphological and semantic relations. For example, occurrences of different lexical units in the same context implies the relation of synonymy. Furthermore, words that are inflectional variants of each other are expected to share certain semantic properties. Table 3[1] is an example of inflectional relation and Table 4[2] of derivational relation.

[1] Extracted from the corpus of *Management* scientific journals I used for the evaluation of EXALP.

[2] *Idem*

Table 9: Example of inflictional relation extracted from bilingual data

English	*Serbian*	*English*	*Serbian*
results	rezultati	try	pokuša
	rezultate		pokušajmo
	rezultata		pokušavaju
	rezultatima		pokušate

Table 10: Example of derivational relation extracted from bilingual data

English	*Serbian*	*English*	*Serbian*
supply	snabdevanje (N)	permit	dozvola (N)
	nabavka (N)		dozvoljene (Adj)
	snabdeti (V)		dozvoliti (V)
	nabaviti (V)		

5.1.3 Terminology extraction

A specific case of building a dictionary is extracting the terminology. The case differs because new terminology constantly emerges and it is very important to do the extraction efficiently and within a small time frame, so the ontologies can keep up to date with the changes. Consistency and the usage of the terms needs to be assured, especially in the political or technical domain [57]. This is explained in more detail in section 2.2.2. on the example of terminology extraction I did during my internship.

5.2 Translation

5.2.1 Machine translation

Machine translation systems are based on probabilistic translation models [12]. These systems are trained on parallel corpora which are sentence aligned. Based on this model, the task of statistical machine translation is to pick a source sentence, that is the most probable translation of a given sentence [15]. If there is enough parallel data at our disposal, it is possible to count how many times a word, phrase or structure is mapped to each of its possible translations [28].

The probabilistic approach has been present in the machine translation systems for many years, however, in the recent years we have seen great emergence of machine translation trained on neural networks. These systems have shown better results for certain pairs of languages than statistical MT systems [30]. The training of neural networks is done on such parallel data. By recognizing different patterns, i.e. the ways certain words are translated in which contexts, neural networks are able to capture layers beyond simple statistics and thus yield better results.

5.2.2 Improving human translation

Parallel data can provide a lot of insights in the domain of translation studies. The author in [41] points out several advantages of using parallel data, claiming that parallel corpus can reveal characteristics of certains texts like tendencies towards explicitness and avoidance of repetition. Furthermore, it is possible to extract the way certain phrases are often translated and search the source of translation errors enabling well explained solutions, which will bring higher quality, uniformed translations in the long run.

5.2.3 Translation memories

In the last decade translation oriented applications have reached significant success because of the emergence of vast amount of data in machine readable format and overall greater computational resources [43] [27]. The idea behind the translation memories is to reuse the text segments which have already been translated in the past and incorporate them in software which aids human translation, making the translating process more efficient. In fact, translation memories can be considered as a speacial case of parallel texts which are ogranized in a database to be used by translators. The extracted bilingual data of high level confidence can be added to these memories, therefore significantly expanding the databases. Moreover, this leads to a more uniform style of translating and adopting new terminology.

5.3 Second language acquisition

Textual corpora in general has been established as a successful teaching aid in the classroom. The corpus provides the students with actual occurrences of language items which help them acquire them more efficiently. They are able to make certain hypotheses and test them on real data [74]. The great advantage of having parallel texts in this context is that an enormous amount of content becomes instantly comprehensible. Aligned corpora are generally used in conciousness-raising exercises. Learners work with language elements such as modals, prepositions, conjuncts or pronouns which are presented in both source and target language. They are asked to reflect on the differences and similarities between items and come to independent conclusions.

Despite these great advantages, it is quite difficult to find adequate parallel bilingual texts that would appeal to language learners, even in the published form. Most of them are 19th century classics and modern literature is completely neglected. In this context, EXALP can create fairly accurate bilingual texts, if the user can provide the source and target texts. The additional advantage of this is that the choice is completely dependent on the user. In fact, the alignment of literary texts seems to have the most accurate alignment on the surface level, however, as I do not posses the aligned data, it was impossible to quantify the results.

5.4 Contrastive linguistics research

Contrastive analysis in linguistics is mainly concerned with bilateral language comparisons [31]. The choice of the pair of languages vary depending on the applications behind it. The comparisons may be between native and foreign language, first and second language, source and target language.

Parallel texts are able to give insights in languages compared which are unlikely to be noticed in monolingual data. Additionally, they can be used for all kinds of comparative purposes which will lead to a better understanding of cultural, typological and language-specific differences and universal features. Moreover, verifying hypotheses against representative data and obtaining quantitative results on the bases of frequency and distribution yields in reliable and complete contrastive research [26].

Chapter 6

Conclusion

The objective of this thesis was to facilitate and enhance the process of creating sentence aligned Serbian-English corpora. This was achieved through extensive research and the development of `EXALP`.

During the internship, I had an opportunity to get a detailed overview of each step which is now present in the pipeline. Additionally, steps that were performed by other people, like raw text extraction and correction, are included, as well. This experience was valuable for further development, as it allowed me to be aware of all special cases that might occur with the text processing and the alignment.

In the introductory part of thesis, the importance of sentence aligned corpora was stressed, as well as their lack in low-resource languages, specifically Serbian. A tool like `EXALP` is able to process large, unstructured and noisy data in order to respond to urgent need of resource gathering. Subsequently, other branches of NLP would be able to exploit this data and improve the status of Serbian in the digital form.

The structure of `EXALP` was partially motivated by the processes of compilation of other parallel corpora and additional research of currently available open-source resources. In-depth research into the sentence alignment algorithms gave me a better understanding of the limitations and advantages of using automatic systems to achieve the alignment, as well as their possible reach and performance expectancy. I opted to implement `hunalign` and `Gargantua` algorithms because of their popularity, ease of implementation and most importantly their ability to give good performance on a language without significant lexical resources.

The development of `EXALP` was a complex task and its main challenge was to connect all the steps in the pipeline so that from each step, little to no errors would propagate to the following one. As it is usually the case with processing unstructured texts, special cases which require handling tend to emerge with every dataset. Most of these special cases were related to the format, layout or structure of the text. Language related cases were mostly handled for Serbian. The alignment algorithms `hunalign` and `Gargantua` are applied to the extracted data and their output is presented in the user-friendly readable format. Along the way, the outputs of all steps are saved which allows for error control or use in other applications.

The evaluation of `EXALP` against manually compiled data, showed that `hunalign` performs better than `Gargantua` in the case of English and Serbian, and in this environment. Additionally, a more detailed analysis of the options `hunalign` uses showed that the dictionary is not paramount for satisfactory performance, which makes it convenient for a low-resource language.

The evaluation phase required an additional script to adequately measure the precision and recall against manually extracted and aligned data. The results showed that on average, `EXALP` has a success rate of 84%. This is a satisfactory outcome, considering all the previous work was done completely manually.

Finally, EXALP's output can have many possible applications in the domain of NLP and linguistic research, as shown in the previous section.

6.1 Future work and limitations

The evaluation phase showed that the recall measure has room for improvement. More specifically, it means that the EXALP missed the extraction of roughly 20% of sentences that occurred in the manually compiled dataset. Considering that this thesis was primarily concerned with the pairing of sentences, it is no surprise that the extraction phase has room for improvement. In this regard, I would say time is the main limitation, as this is an area completely independent of sentence alignment. More successful text processing is possible with further analysis of errors and research into this domain. Nevertheless, I believe that the extraction phase in EXALP makes a good starting point for further development. Big advantage of this part of EXALP is that it is completely language independent, therefore can be applied to any pair of languages. Additionally, various formats of unstructured data can easily be added.

In the sentence delimitation phase, it is possible to easily add other languages, more precisely any language which is supported by Unitex.

Finally, the evaluation showed that the alignment has the precision of 94% on average, which is more than satisfactory. Additionally, the improvement of the text processing phase would inevitably improve the precision rate, as well.

In the future, EXALP can be available online, with a complete user interface. The user would have an option of uploading two bilingual files which would be processed and give a user-friendly output. Additionally, it would be good to add options of easily manually correcting the data. In this way, EXALP would be of good use not only in the industry or academia, but for personal use, as well.

I believe that with some additional development and work on text extraction, EXALP can find its place in the compilation of Serbian-English corpora. However, even with 84% accuracy, its use saves time and effort through automatization of processes otherwise done by several people and spread over significant amount of time.

Bibliography

[1] Bibliša - documentation. http://jerteh.rs/biblisha/Documentation.aspx/. Accessed: 2010-11-04.

[2] A brief history of word embeddings (and some clarifications). https://www.gavagai.se/blog/2015/09/30/a-brief-history-of-word-embeddings. Accessed: 2017-04-25.

[3] Extensive markup language. https://en.wikipedia.org/wiki/XML. Accessed: 2010-11-07.

[4] JeRTex. http://jerteh.rs/?page_id=84&lang=en. Accessed: 2016-10-30.

[5] LeXimir. http://korpus.matf.bg.ac.rs/soft/LeXimir.html/. Accessed: 2016-11-02.

[6] META-NET. http://www.meta-net.eu/. Accessed: 2016-11-02.

[7] Xml tools plugin. https://sourceforge.net/projects/npp-plugins/files/XML%20Tools/. Accessed: 2010-11-14.

[8] Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel cor-

pora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604. Association for Computational Linguistics, 2005.

[9] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[10] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[11] Fabienne Braune and Alexander Fraser. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 81–89. Association for Computational Linguistics, 2010.

[12] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[13] Roberto Busa. The annals of humanities computing: The index thomisticus. *Computers and the Humanities*, 14(2):83–90, 1980.

[14] Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar F Zaidan. Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53. Association for Computational Linguistics, 2010.

[15] Chris Callison-Burch, David Talbot, and Miles Osborne. Statistical machine translation with word-and sentence-aligned parallel corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 175. Association for Computational Linguistics, 2004.

[16] Stanley F Chen. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 9–16. Association for Computational Linguistics, 1993.

[17] Pernilla Danielsson and Daniel Ridings. Practical presentation of a “vanilla” aligner. In *Proceedings of the TELRI Workshop on Alignment and Exploitation of Texts*, 1997.

[18] Wim De Mulder, Steven Bethard, and Marie-Francine Moens. A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech & Language*, 30(1):61–98, 2015.

[19] Andreas Eisele and Yu Chen. Multiun: A multilingual corpus from united nation documents. In *LREC*, 2010.

[20] Tomaz Erjavec. Multext-east version 3: Multilingual morphosyntactic specifications, lexicons and corpora. In *LREC*, 2004.

[21] W Nelson Francis and Henry Kucera. Brown corpus manual. *Brown University*, 2, 1979.

[22] William A Gale and Kenneth W Church. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102, 1993.

[23] Roger Garside, Geoffrey N Leech, and Tony McEnery. *Corpus annotation: linguistic information from computer text corpora.* Taylor & Francis, 1997.

[24] Raymond G Gordon, Barbara F Grimes, et al. *Ethnologue: Languages of the world*, volume 15. sil International Dallas, TX, 2005.

[25] Robert D Greenberg. *Language and identity in the Balkans: Serbo-Croatian and its disintegration*. OUP Oxford, 2004.

[26] Hilde Hasselgård. *Information structure in a cross-linguistic perspective*. Number 39. Rodopi, 2002.

[27] Matthias Heyn. Integrating machine translation into translation memory systems. In *Proceedings of the EAMT Machine Translation Workshop, TKE'96*, pages 113–126, 1996.

[28] Dan Jurafsky. *Speech & language processing*. Pearson Education India, 2000.

[29] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86, 2005.

[30] Nenad Koncar and Gregory Guthrie. A natural language translation neural network. In *New Methods in Language Processing*, pages 219–228, 1997.

[31] Ekkehard König. The place of contrastive linguistics in language comparison. *Languages in Contrast*, 2011.

[32] Andreas Koutsoudas and Assya Humecky. Ambiguity of syntactic function resolved by linear context. *Word*, 13(3):403–414, 1957.

[33] Wessel Kraaij, Jian-Yun Nie, and Michel Simard. Embedding web-based statistical translation models in cross-language information retrieval. *Computational Linguistics*, 29(3):381–419, 2003.

[34] Cvetana Krstev and Duško Vitas. An aligned english-serbian corpus. *ELLSIIR Proceedings (English Language and Literature Studies: Image, Identity, Reality)*, 1:495–508, 2011.

[35] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

[36] Geoffrey Neil Leech. 100 million words of english: the british national corpus (bnc). 1992.

[37] Els Lefever, Lieve Macken, and Veronique Hoste. Language-independent bilingual terminology extraction from a multilingual parallel corpus. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 496–504. Association for Computational Linguistics, 2009.

[38] Omer Levy, Anders Søgaard, Yoav Goldberg, and Israel Ramat-Gan. A strong baseline for learning cross-lingual word embeddings from sentence alignments. *arXiv preprint arXiv:1608.05426*, 2016.

[39] Peng Li, Maosong Sun, and Ping Xue. Fast-champollion: a fast and robust sentence alignment algorithm. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 710–718. Association for Computational Linguistics, 2010.

[40] Xiaoyi Ma. Champollion: A robust parallel text sentence aligner. In *LREC 2006: Fifth International Conference on Language Resources and Evaluation*, pages 489–492, 2006.

[41] Kirsten Malmkjær. Love thy neighbour: Will parallel corpora endear linguists to-translators? *Meta: journal des traducteurs/Meta: Translators' Journal*, 43(4):534–541, 1998.

[42] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[43] Hiroshi Masuichi, Michihiro Tamune, Masatoshi Tagawa, Kiyoshi Tashiro, Atsushi Itoh, Kyosuke Ishikawa, and Naoko Sato. Translation memory system, September 7 2005. US Patent App. 11/219,660.

[44] I Dan Melamed. A geometric approach to mapping bitext correspondence. *arXiv preprint cmp-lg/9609009*, 1996.

[45] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[46] Ruslan Mitkov. *The Oxford handbook of computational linguistics*. Oxford University Press, 2005.

[47] Robert Moore. Fast and accurate sentence alignment of bilingual corpora. *Machine Translation: From Research to Real Users*, pages 135–144, 2002.

[48] Éva Mújdricza-Maydt, Huiqin Körkel-Qu, Stefan Riezler, and Sebastian Padó. High-precision sentence alignment by bootstrapping from wood standard annotations. *The Prague Bulletin of Mathematical Linguistics*, 99:5–16, 2013.

[49] Sébastien Paumier. Manuel d'utilisation du logiciel unitex. *Université de Marne-la-Vallée*, 2002.

[50] Sébastien Paumier. Unitex-manuel d'utilisation. 2011.

[51] Randolph Quirk. The survey of english usage. *Transactions of the Philological Society*, pages 70–87, 1960.

[52] Adwait Ratnaparkhi. Mxterminator, 1997.

[53] Ulrich Schafer. Standard xml query languages for natural language processing. 2009.

[54] Michel Simard, George F Foster, and Pierre Isabelle. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing-Volume 2*, pages 1071–1082. IBM Press, 1993.

[55] Ranka Stanković, Cvetana Krstev, Duško Vitas, Nikola Vulović, and Olivera Kitanović. Keyword-based search on bilingual digital libraries. In *Proceedings of the 2nd International KEYSTONE Conference,(Cluj-Napoca, Romania*, 2016.

[56] Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. *arXiv preprint cs/0609058*, 2006.

[57] Della Summers. Corpus lexicography–the importance of representativeness in relation to frequency. *Longman Language Review*, 3:6–9, 1996.

[58] Jan Svartvik and Randolph Quirk. *A corpus of English conversation*, volume 56. Studentlitteratur, 1980.

[59] Wolfgang Teubert. The role of parallel corpora in translation and multilingual lexicography. *Lexis in contrast*, pages 189–214, 2002.

[60] Samuel Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky. Deep neural network features and semi-supervised training for low resource speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6704–6708. IEEE, 2013.

[61] Jörg Tiedemann. Parallel data, tools and interfaces in opus. In *LREC*, volume 2012, pages 2214–2218, 2012.

[62] Jörg Tiedemann and Lars Nygaard. The opus corpus-parallel and free: http://logos. uio. no/opus. In *LREC*, 2004.

[63] Miloš Utvić, Ranka Stanković, and Ivan Obradović. Integrisano okruženje za pripremu paralelizovanog korpusa. *Die Unterschiede zwischen dem Bosnischen/-Bosniakischen, Kroatischen und Serbischen*, pages 563–578, 2008.

[64] Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. Parallel corpora for medium density languages. *AMSTERDAM STUDIES IN THE THEORY AND HISTORY OF LINGUISTIC SCIENCE SERIES 4*, 292:247, 2007.

[65] Jean Véronis. *Parallel Text Processing: Alignment and use of translation corpora*, volume 13. Springer Science & Business Media, 2013.

[66] Duško Vitas, C Krstev, I Obradović, Lj Popović, and Gordana Pavlović-Lažetić. Processing serbian written texts: an overview of resources and basic tools. In *Workshop on Balkan Language Resources and Tools*, volume 21, pages 97–104, 2003.

[67] Duško Vitas and Cvetana Krstev. Literature and aligned texts. *Readings in Multilinguality*, pages 148–155, 2006.

[68] Dusko Vitas, Cvetana Krstev, and Eric Laporte. Preparation and exploitation of bilingual texts. *Lux Coreana*, 1:110–132, 2006.

[69] Duško Vitas, Ljubomir Popović, Cvetana Krstev, Ivan Obradović, Gordana Pavlović-Lažetić, and Mladen Stanojević. *The Serbian Language in the Digital Age*. META-NET White Paper Series. Georg Rehm and Hans Uszkoreit (Series Editors). Springer, 2012. Available online at `http://www.meta-net.eu/whitepapers`.

[70] Merriam Webster. Merriam-webster online dictionary. 2006.

[71] Jinxi Xu, Ralph Weischedel, and Chanh Nguyen. Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 105–110. ACM, 2001.

[72] Yong Xu, Aurélien Max, and François Yvon. Sentence alignment for literary texts. *LiLT (Linguistic Issues in Language Technology)*, 12, 2015.

[73] Qian Yu, Aurélien Max, and François Yvon. Revisiting sentence alignment algorithms for alignment visualization and evaluation. In *The 5th Workshop on Building and Using Comparable Corpora*, page 10, 2012.

[74] Federico Zanettin. Parallel words: Designing a bilingual database for translation activities. *Corpora in language education and research: a selection of papers from Talc*, 94:99–111, 1994.

[75] Ying Zhang, Ke Wu, Jianfeng Gao, and Phil Vines. Automatic acquisition of chinese-english parallel corpus from the web. In *ECIR*, volume 3936, pages 420–431. Springer, 2006.

Place Blaise Pascal, 1 bte L3.03.11, 1348 Louvain-la-Neuve, Belgique www.uclouvain.be/fial