# NLP methods for the automatic generation of exercises for second language learning from parallel corpus data

Arianna Zanetti

Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg

Supervisors:
Elena Volodina
Johannes Graën

UNIVERSITY OF GOTHENBURG

NLP methods for the automatic generation of exercises for second language learning from parallel corpus data

guszanear@student.gu.se

To my friends and family

# NLP methods for the automatic generation of exercises for second language learning from parallel corpus data

Arianna Zanetti

Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
Gothenburg, Sweden

## ABSTRACT

Intelligent Computer Assisted Language Learning (ICALL), or Intelligent Computer Assisted Language Instruction (ICALI), is a field of research that combines Artificial Intelligence and Computer Assisted Language Learning (CALL) in order to produce tools that can aid second language learners without human intervention.

The automatic generation of exercises for language learners from a corpus enables the students to self-pace learning activities and offers a theoretically infinite, un-mediated and un-biased content.

In recent years, the advancement in NLP technology and the increase of available resources made this possibility closer. In particular, relevant sources of knowledge are the large collections of aligned parallel texts: corpora containing sentences in different languages, which can be considered translations of one another.

The present work explores the possibility to extract candidate sentences and their translations from a parallel corpus and use them to generate exercises for different proficiency levels.

The research was conducted experimenting with several available NLP tools and qualitatively evaluating the results on a training set of documents to define

a pipeline for the language pairs: Swedish-English, English-Italian, Swedish-Italian. Finally, a set of 30 random documents was extracted and annotated manually to obtain a quantitative evaluation. The results showed a mean accuracy between 70-90% in the sentence selection, depending on the language pair; between 80-96% using more strict criteria for the selection and reducing the recall.

It is interesting to note that the implementation is mostly language independent, there is only one language-specific component to estimate the target proficiency level of the sentence, so in future works the same pipeline could be extended to include other language pairs.

**Keywords**: ICALL, language learning, parallel corpus, exercise generation

# CONTENT

# INDEX OF FIGURES

# INDEX OF TABLES

# ABBREVIATIONS

| | |
|---|---|
| CALL | Computer Assisted Language Learning |
| CEFR | Common European Framework of Reference |
| CQL | Corpus Query Language |
| CQP | Corpus Query Processor |
| DDL | Data-Driven Learning |
| DEPREL | Dependency Relation |
| EM | Expectation Maximization |
| GBL | Game Based Learning |
| GDEx | Good Dictionary Examples |
| GNU | GNU's Not Unix |
| HMM | Hidden Markov Model |
| ICALI | Intelligent Computer Assisted Language Instruction |
| ICALL | Intelligent Computer Assisted Language Learning |
| L2 | Second Language |
| LSP | Language for Specific Purpose |
| MCMC | Markov Chain Monte Carlo |
| MT | Machine Translation |
| NLP | Natural Language Processing |
| PARSNIP | Politics Alcohol Religion Sex Narcotics Isms Pork |
| POS | Part Of Speech |
| SLA | Second Language Acquisition |
| SSH | Secure Shell |
| TM | Translation Memory |
| TTR | Type-Token Ratio |
| UTF-8 | Unicode Transformation Format 8 bit |
| XCES | XML Corpus Encoding Standard |

# 1 INTRODUCTION

Teubert (1996) and Lawson (2001) state that a parallel corpus of reasonable size contains more knowledge for a learner than any bilingual dictionary. At the present day, the enormous diffusion of the web made it possible to create corpora and parallel corpora for more than 100 languages, containing billions of tokens or parallel units. Despite this, the research in using data from a parallel corpus for language learning is still at the beginning. This gives the motivation for this work.

The general research question can be expressed as: how can natural language processing be applied on a parallel corpus for second language learning?

To explore it, it is necessary to separate the question in two sub-parts:

1. Is it possible to extract candidate sentences from a parallel corpus to generate exercises for second language learning without human intervention?

2. Which pre-processing operations are necessary?

After studying the background of the fields of ICALL and parallel corpora, the research was conducted experimenting with several available NLP tools and qualitatively evaluating the results on a training set of documents, to define a pipeline for the language pairs: Swedish-English, English-Italian, Swedish-Italian. Finally, the pipeline was implemented and tested on a set of 30 random documents extracted from the corpus and annotated manually to obtain a quantitative evaluation.

The presentation continues as follows:

- The first chapter summarizes the state-of-the-art and supports the pedagogical motivation behind the automatic generation of exercises.

- The second chapter introduces the methodology and describes the different tools used for the implementation and test, other than presenting an example of a novel exercise type.

- The third chapter gives more details about the pipeline defined after the qualitative tests, explains the technical aspects of the implementation of the different components and shows the results obtained for each sub-part of the project.

- Finally, the fourth chapter summarizes the conclusions and the fifth discusses the future work.

# 2 STATE OF THE ART

## 2.1 ICALL

Intelligent Computer Assisted Language Learning (ICALL), or Intelligent Computer Assisted Language Instruction (ICALI), is a field of research that combines Artificial Intelligence and Computer Assisted Language Learning (CALL) in order to produce tools that can aid second language learners without human intervention. This is very useful because, as Dodigovic (2005) writes, "a machine able to understand natural language would be a tireless language model, interlocutor and error-correcting and grading authority". The possibility to practice a non-native language with a software has proved to be beneficial for many reasons: it enables the student to self-pace learning activities, it capitalizes on the fascination many people find in computers, allows easy access to the enormous potential of the web resources and offers an environment where the learner can feel safer in making a mistake. From a pedagogical point of view, for many people the possibility to test their knowledge in a more flexible way and training as many times as they want on a subject without the risk of being embarrassed in front of a colleague or teacher would result in quicker and better improvements in their skills. Moreover, the system could keep track of the user's progress to provide more specific feedback, taking into account, for example, their mother tongue to predict positive or negative L1 transfer, or previous answers they entered, to tailor the teaching activities to the specific needs and guide the learning process.

This is, of course, an ideal representation. There are still many issues to consider, the most important one being the reliability of computers and software technology. It is one thing to give feedback when the set of possible answers is predictable and limited to a restricted domain, another to offer a general and dynamic tool the students can trust to progress to further stages of independence. Salaberry (1996) urges caution in applying Natural Language Processing (NLP) to CALL applications, because it cannot account for the full complexity of natural human languages, and a similar negative perception seems to be shared among CALL developers (Nerbonne, 2002). It is true that there are many challenges in processing learners' language, especially because most of the NLP tools are built for correct language and do not support non-native speakers' errors. For this reason, most grammar checkers today are built expecting the users to have enough linguistic intuition to critically evaluate the

responses. These limitations must be taken into consideration when designing and implementing a CALL software and not only developers, but also learners using the software must be aware of them (Higgins, 1987).

Other objections to ICALL relate to pedagogical issues. Mishan and Strunz (2003) state that NLP tools represent a "solution in search of a problem", meaning they are only technology-driven and interested in what computers can or cannot do, instead of considering the real linguistic and methodological implications of a program. Oxford (1993), summarizing the situation in the mid-1990s, criticizes the excessive attention to technology at the expense of language learning/teaching principles. Potter (2004) argues that the risk with automatic learner assessment is to lose the "humanist approach" that cares more about creativity and communicating meaning than punctuation and grammar and, especially in essays grading, it could lead to students artificially inflating their writing with complex linguistic structures even when it is not appropriate to mislead the system and obtain a better grade.

Finally, few systems consider the profound difference between written and spoken language, which is a fundamental aspect of language learning.

It is also nontrivial to evaluate an ICALL system, because the software structure is not transparent, it is often difficult to understand its content and operativity, and an example-based assessment requires time and effort to prepare the data and there is a high risk of having biases that prevent the results from being accurate. The optimal solution would be to have a software for language learning developed jointly by experts of the different fields involved: language specialists, psychologists and language pedagogists, teachers and computer scientists. This is a unique challenge, because most teachers have a limited experience in using technology and usually the perception of the purpose is different between them and the programmers.

Despite this technological and ideologic gap between research in NLP and courseware development, the interest in more linguistic aware applications is shared among experts of the different domains, as shown by the different conferences organized on the topic, the first one being Applied Natural Language Processing in 1997 who devoted a session to CALL, and by the increasing proportion of articles in journals dedicated to CALL (e.g., CALL, ReCALL, CALICO). Even if a computer cannot completely substitute the human being in what is called a "tutor" application, designed to imitate the functions of a teacher, it is possible to develop ICALL systems at different levels, realizing "tools" for specific tasks (Levy, 1997), like analysing syntactic and/or semantic correctness of the user inputs.

It is fundamental to consider pedagogical and linguistics aspects in the need analysis stage of the development, before moving to the design, implementation, and evaluation.

For the system to be effective, the material to which learners are exposed must be comprehensible, and feedback appropriate to their current proficiency level (Van del Linden, 1993) and to the learner's type (Suphawat, 1999). For example, communicative learners will prefer learning by examples, so they will want the system to correct their mistakes, while analytical learners will prefer a brief and schematic indication of an error, to have the chance to read more about the specific topic and try to solve it themselves. This suggests it would be advisable to have a model of the teaching activity, to know which topics and constructs are suitable at each level, and a model of the student, in order for the technology to remain neutral and give every user the chance to benefit from it, keeping track of their features but also their progresses.

### 2.1.1 NLP TECHNIQUES FOR ICALL

Nerbonne (2002) identifies four main NLP contributions towards the implementation of CALL software – other than speech recognition, which lies outside of the present discussion:

- Concordancing, concordance programs to find keywords in their context. Concordancing and lemmatization provide easier and more flexible access to authentic language data, collected in the form of a corpus.

- Text alignment, alignment of bilingual documents. Bilingual corpora, created with text alignment, provide the translation in a known language along with the examples of real language use. It cannot be used as a proper translation, but the texts are assumed to have pragmatic equivalence, therefore the comparison can make the text in the foreign language easier to understand for the learner.

- Morphological processing, lemmatization and morphological generation. Morphological and syntactic processing are used to create new exercise material and compare the sentences written by the students. Traditionally, work on language

generation has been applied to create games (Pasero and Sabatier, 1998), or simple dialogues (Hamburger, 1995).

- Syntactic processing, parsing to clarify the linguistic structure of a sentence and diagnose irregularities in the learners' output.

Parsing is, undoubtedly, the most important component of NLP-based CALL applications, to the extent that many researchers suggest the name Intelligent CALL should be changed to "parser-based CALL" (Heift and Schulze, 2007). According to Matthews (1993) at the time there were still too few programs that incorporated parsers, because the technique, along with their knowledge about linguistics, had only recently reached a level that permitted fully functional tools and the development of a parser with computational grammars and its integration in a package were still complex, time-consuming and expensive. A similar observation is also made by Dodigovic (2005), even if she argues that the reason is not the lack of technological advancement but the limited cooperation between experts of the fields involved.

A broader application of parsers would enable the computer to encode complex grammatical knowledge and recognize the students' mistakes and react to them in a more sophisticated way than simple keywords or pattern matching. The problem is that sentences tend to be syntactically ambiguous, so even if it is possible to use a parser to look for irregular grammatical structures, the system may find a possible correct interpretation even if it was not the intended one. Other than that, if the parser does not find any plausible analysis, it is not immediate to understand the source of the error and thus give meaningful feedback about it or correct it. This is especially true when parsers are implemented using machine learning, because these techniques "learn" from examples in a way that is not transparent to the programmers, so it is not possible to determine why a particular outcome is obtained.

One solution is to anticipate the different sources of errors, and create models of the "students' grammar", with the aid of the so-called *mal-rules* (Matthews & Fox, 1991), rules which are not part of the grammar of the language but explicitly cover the students' mistakes. The problem with this approach is that an unanticipated error goes undetected and the cause of the error is not accounted for. A more robust diagnose procedure can be obtained using a large learner corpus, with material gathered from genuine communication by non-native speakers of the language, but such corpus is not easy to build because learner language is influenced by linguistic, situational and psycholinguistic factors and a lack of control over these different features results in unreliable

findings (Granger, 2008). First, often in second language learning the source of errors is inter-lingual, i.e. caused by the interference of the first language grammar, so if native speakers of a language are excluded from the corpus, their specific mistakes might not be considered, no matter how frequent they might be. Even more complicated is to predict the errors when they are not caused by the transfer between native and target language but between a previously learnt language and the one currently being studied. In this case, the language used by the student lies somewhere between all three (or possibly more) languages.

The matter of which errors to consider and how to give feedback has been source of debate among researchers in ICALL. While most of them agree that reporting all the errors at the same time overwhelms the students, especially at lower proficiency levels, one line of thought suggests to keep a priority queue and select the most appropriate one to correct, based on the frequency and importance of the error and on the student's proficiency level, in a schematic, more computer-oriented way (Van der Linden, 1993). The other to recreate a typical language acquisition environment, in which the system gives a more conversational response, acknowledging both the presence of the mistake and the correction at the same time with the use of recast[1] (Reeder et al., 1999).

### 2.1.2 EXERCISE GENERATION

Many teachers prefer to create their own material so that sentences used in examples or exercises can be contextualized to the specific social and cultural context (Pilán, 2013), but manual generation is a time-consuming task. Real sentences from a corpus allow the students to practice with theoretically infinite, general, un-mediated and un-biased content. Cobb and Boulton (2015) found 116 empirical studies to support the benefits of data-driven learning. According to those, the exposure to authentic input is both effective and efficient in supporting intuition and helping the learners gradually reproduce the underlying lexical, grammatical, pragmatic, and other patterns implicit in the language they encounter.

[1] Techniques used in language teaching to correct learners' errors in such a way that communication is not obstructed. To recast an error, an interlocutor will repeat the error back to the learner in a corrected form. Recasts are used both by teachers in formal educational settings, and by interlocutors in naturalistic language acquisition.

Of course, not any sentence can be used for this purpose. It is necessary to select appropriate examples, depending on the type of exercise and the proficiency level of the learner. The sentence must be well-formed, comprehensible to the students in style, register and vocabulary, and sufficiently context-independent, but there are no agreed characteristics that make a "good sentence". Too many constraints could result in over-simplified sentences, while out-of-the-ordinary examples could be more interesting or relate to the learners' world of experience and be better from a pedagogical point of view (Segler, 2007).

A large part of the research in this area comes from the selection of good dictionary examples (**GDEx**; Kilgarriff et al., 2008), because in both cases the sentence must be readable out of context, lexically and structurally.

There are two general methods (Pilán et al., 2013):

- Machine learning
- Natural Language Processing

Machine learning techniques are based on human-annotated sentences and estimate the parameters of a model using multiple linguistics dimensions at the same time. It can be an important step in determining which features are the most predictive and help labelling the complexity of a sentence because, most of the times, the criteria teachers use to select them cannot be verbalized, they derive from intuition.

Natural Language Processing techniques, on the other hand, are based on pre-defined rules written by specialists and are thus more customizable and can be suitable for a wider range of applications.

In reading comprehension tasks, it is more important to consider vocabulary knowledge. Second Language Acquisition (SLA) research has established that the student should know at least 95-98% of the words in a text to comprehend it (Laufer and Ravenhorst-Kalovski, 2010). When the example's goal is to demonstrate the use of a word, it is important to reduce the syntactic complexity of the sentence, and to use semantically related words and frequent co-occurrences (Segler, 2007), but also to differentiate the use of the word in different contexts (Minack et al., 2011).

An example of a tool built considering all these different elements is **HitEx** (Pilán et al., 2016). It uses machine learning to measure the proficiency level needed to understand a sentence, and customizable criteria to look for the

presence or absence of linguistic elements, in particular: finite verb and subject, punctuation, absence of a large amount of non-alphabetical tokens, absence of connectives in initial position and anaphoric expressions, negative wording, direct speech, sensitive vocabulary, etc.

For language learning exercises, it is particularly important to assign a target proficiency level to a sentence, using complexity or readability measures. These can be based on the raw text, considering the number and length of the words (Kincaid et al., 1975), or exploit NLP features: syntax (Schwarm and Ostendorf, 2005), and discourse (Barzilay and Lapata, 2008). The first ones are less advanced but have the advantage to be language independent, while the others can be difficult to transfer from a language to another.

Machine learning methods consider the attribution of a proficiency level to the sentences as a classification task, other than the already mentioned HitEx, a similar model is used in **READ-IT** (Dell'Orletta et al., 2011) and **CEFRLex** (Dürlich and François, 2018). READ-IT is based on two different corpora from newspapers article to separate the sentences in easy or difficult. CEFRLex relies on pedagogical materials tagged with one of the six levels of language proficiency internationally accepted as standard (**CEFR**): A1 (beginner), A2 (elementary), B1 (intermediate), B2 (upper intermediate), C1 (advanced) and C2 (proficient).

The idea to use the CEFR standard in ICALL to generate exercises for learners with different levels of abilities is not new. The framework was created by the Council of Europe to provide guidelines for language teaching and assessment to promote transparency and coherence in language education across languages and countries (Council of Europe, 2001) and it is widely accepted by human teachers. The problem is its flexibility. The competence and skills needed at any level are not clearly defined and there is room for interpretations in different languages and target groups (Volodina et al., 2013). Also, the description of the levels is based more on the teachers' perceptions than on empirical evidence from learners' data.

A possibility is to use pedagogical materials already created by teachers as a training corpus for the different levels, but it is not necessarily effective, because what separates one level to the next is the topics students are expected to be able to discuss, not "easy" or "complex" words that appear more or less frequently. This implies that even more complex or less frequent words can be comprehended earlier if they are needed to talk about a subject.

## 2.2 PARALLEL CORPORA

Corpora are an essential source of information for any type of linguistic research or statistic-based system. Being a finite collection of spoken or written utterances of natural language, they can never be complete, but they can be representative of the language in some respect, for example the learner corpora (see section 2.1.1) used to analyse the most frequent grammar mistakes made by a group of students; corpora which document a specific genre (LSP); bilingual/multilingual corpora, to study interlinguistic phenomena and find examples of previous translations of a sentence in different languages.

The use of corpora has a very long history in linguistics, translation, and language learning. Examples of previous translations of a sentence have always been used by human translators to learn and improve, gaining more insight into the use of words or expressions in context. According to Leech (1997) the first conference program referring the use of corpora in language teaching was in 1992. Already in the 90's, Johns and King (1991) created the term "data-driven learning" (DDL) to emphasize the role of tools for corpus analysis in a language classroom.

Corpus data can be useful in CALL in different ways. Advanced students or teachers can search specific examples to explain the nuances of meaning, common usages of a term, or stylistic differences; at lower proficiency levels, learners can discover the linguistic data and gain a better understanding of grammar through examples. In a more indirect way, corpora can be used to compile dictionaries or frequency lists, or for textbook materials.

Parallel corpora are aligned bilingual corpora. The first experiments to use them in natural language processing were in the late fifties, with not very encouraging results caused by the limitations of the computers of the time and the reduced availability of textual data in digital format (Santos, 2011).

In 1970s, a new promising form of parallel corpus called Translation Memory (TM) was introduced[2]. At the same time, the interest in aligned text was renewed due to the increasing amount of large data.

TMs are systems, usually integrated in a word-processing environment, which store all the sentences translated by human translators. Being written by a human expert, the sentences are expected to be grammatically correct and the alignment of the TM units can be assumed to be accurate. When a segment that

[2] https://www.sdltrados.com/solutions/translation-memory

has already been translated and saved is encountered a second time in the text, the TM retrieves its translation and gives it as a suggestion to the user. Systems of this type are still incorporated in online translation services like Google Translate, to help "pure" machine translation.

Other than offering translations, or translation support, an aligned parallel corpus can be applied to extract bilingual lexicons, or to discover morphological and semantic relations, exploring the patterns of the extracted matches. This can be especially helpful for under-resourced languages in which a monolingual corpus does not have enough data. For example, Graën et al. (2020) hypothesize that if similar expressions in two languages are strong translations, i.e. they are frequently translated with each other, they have similar CEFR levels, so the model trained on a bigger monolingual corpus could be employed to assess the CEFR level of the aligned documents.

The majority of systems are built only for the English language, especially because of the attention English has received in formal linguistics which offers the prerequisite for a more successful work in building computational grammars, but also because of the high demands of resources for English. Therefore, the possibility to transfer some of this knowledge to a different language would result in a huge advancement in the field.

The crucial part is the alignment. Corresponding segments, usually sentences or paragraphs, need to be matched in order to search for parallel concordances. This allows the user to look for a word or expression in one of the languages and obtain all the sentences that have been associated to it in the other, regardless of the direction of the translation.

Typically, there are three layers of alignment: document, sentence, and word. In some cases, it can also be at verse-level (e.g. in the Bible), morpheme, phrase, syntactic constituent, etc. The more coarse-grained levels are performed first, to improve the outcomes of the lower ones. Independently of the type, the most frequent alignment is 1:1, which means the source corresponds exactly to a target text. Less frequently, it can be:

- 1:0, omission
- 0:1, addition
- m:n, usually with $1 \leq m \leq 2$ and $1 \leq n \leq 2$.

Document alignment is very corpus-specific, a document can be an article, a book chapter, or any other unit comprising at least one sentence. Sentence and word alignment, on the other hand, are mostly the same for all corpora. The

first is based on the assumption that the information is expressed in the same order in the documents and it is easier to obtain, with length-based or dictionary-based models, because in the vast majority of cases there are no crossing links and the correspondence is 1:1. The second is a lot more complex. The mapping between words is not monotonic, there are many types of word correspondences, and in the majority of cases it is ambiguous because lexical units are not the same in different languages. It is very common to have partially correct or partially wrong associations, more than complete correctness, due to the nature of translation (Tiedemann, 2003). Usually, the degree of correspondence is expressed as alignment probability and the models are "fuzzy", with no clear-cut separations or strict links.

There are many parallel corpora available, especially for some language pairs. Some of the most common ones come from the transcriptions of Parliament debates in officially bilingual countries such as Canada[3] (English and French), and Belgium[4] (Dutch and French). Large multilingual corpora originate from multinational organizations, like the proceedings of the European Parliament[5], containing 21 European languages, or the United Nations[6], with 6 official languages: Arabic, Chinese, English, French, Russian and Spanish. Other corpora that could be interesting for the number of available languages are compilation of translated books, like the New Testament, which is available in 1001 languages (Östling 2015), but they are not suited for most applications, because they contain less than a million tokens per language.

Other than the data, it is necessary to have effective ways to access it. Simple concordancers can be helpful, but they might not be the best tool for every user group (Volk et al., 2014). Translators are interested in knowing the frequencies of translation variants in different domains and are usually able to identify the corresponding words in the original and target sentences, so they can work with a corpus aligned at the sentence level; learners, on the other hand, may need an alignment at the word level, and they are interested in the typical translations and in the usage of words in context.

A more sophisticated method to access the corpora is the Corpus Query Language (CQL), a code to set criteria for complex searches that may include words or lemmas but also POS tags, sequences or repetitions of tokens, and

[3] https://www.isi.edu/natural-language/download/hansard
[4] https://data.europa.eu/euodp/en/data/dataset/elrc_421
[5] http://www.statmt.org/europarl
[6] https://conferences.unite.un.org/uncorpus

structures. The problem with these types of languages is that they are usually more difficult to understand by the users.

One of the first works on parallel corpus for linguistics was built for the Oslo Multilingual Corpus[7]. It consisted of two tools: the Translation Corpus Aligner (Hofland and Johansson, 1998) and the Translation Corpus Explorer (Ebeling, 1998). They were based on "anchor" words, selected according to their frequency and loaded into a database. An interface allowed basic lexical searches and showed the first sentence where the word was used and the corresponding one in the parallel text.

The majority of tools target translators, especially because word-aligned corpora have become available only in recent years (Bourdaillet et al., 2010), but there are commercial concordancers available through web-services targeted at language learners, like Glosbe[8], Linguee[9], and Tradooit[10], based on online dictionaries with usage examples. Glosbe can be considered a multilingual online dictionary, it contains 125 languages in combination with German but since it focuses more on recall than precision, the words are often mis-aligned. Linguee offers parallel texts in English, French, German, Portuguese and Spanish, plus Chinese, Japanese, Russian, and other languages in combination with English and it shows dictionary entries sorted according to the target words. Tradooit contains 370 million words aligned between English and French and 260 million aligned between English and Spanish. It has an option for the user to rate the alignment to improve the system.

The idea that a parallel corpus can have a significant role in language learning has been explored by many researchers: Teubert (1996), then confirmed by Lawson (2001), state that "a parallel corpus of a reasonable size contains more knowledge for a learner than any bilingual dictionary"; Briscoe and Carroll (2004) show the analysis of a parallel corpus can be applied to more general NLP projects.

Despite the undeniable interest in the field, Heift and Schulze (2007) write that to their knowledge there is no NLP software in CALL that uses a parallel corpus.

[7] https://www.hf.uio.no/ilos/english/services/knowledge-resources/omc
[8] https://en.glosbe.com
[9] https://www.linguee.com
[10] https://www.tradooit.com

# 3 METHODOLOGY AND TOOLS

The goal of this thesis is to investigate how natural language processing can be applied on a parallel corpus for second language learning to automatically generate exercises and give the learners feedback on their solutions. In the previous chapter, it has been argued why the possibility to automatically select sentences from a parallel corpus can be interesting and innovative from an ICALL perspective, both to offer experts and teachers a tool to create didactic material, and to create pedagogically useful exercises that can be solved autonomously. This chapter presents the research method and the different tools used to process and filter the data and it concludes proposing a new type of exercise.

The study was conducted examining example sentences from the chosen corpus and studying the related literature to select which features needed to be extracted. In particular, POS and syntactic dependency labels, sentence and word alignment, and proficiency level needed to understand the words. After that, several pre-processing operations were tested in a qualitative way, to clean and prepare the data, until the pipeline shown in **FIGURE 1** was defined.


*FIGURE 1 – Pipeline*

At every step, the sentences are filtered out if the system determines they are not suitable or if it cannot process them, favouring precision over recall.

## 3.1 DATA

**OPUS** (Tiedemann, 2012) is an open collection of parallel texts from various web resources and domains. The current version contains over 40 billion

tokens in 2.7 billion parallel units and it covers over 90 languages, including otherwise poorly-resourced language pairs like Romanian-Turkish and Bulgarian-Hungarian. The largest domains are legislative and administrative texts and translated movie subtitles, but it also contains data from newspaper texts and other online sources.

Most of the parallel corpora available are from specialized domains, like legislation or administration and do not cover more than two languages (see section 2.2). Movies and TV subtitles, on the other hand, offer a wide variety of genres, from colloquial or slang to narrative - in the case of documentaries - and they are associated to different kinds of valuable information, like the link to a spoken utterance (Tiedemann, 2007). Subtitles are, according to Lison et al. (2018) the world's largest open collection of parallel data. Their conversational nature makes them ideal for exploring language phenomena and properties of everyday language (Paetzold and Specia, 2016). The main disadvantage is that subtitles must obey time and space constraints, and often try to match lip movements whenever talking people are shown in the videos, therefore the translation is freer than in other domains, especially for some language pairs (Lison and Tiedemann, 2016). Sometimes, they summarize the audio instead of completely transcribing it. How they "compress" the conversation differs due to structural divergences between the languages, cultural divergences and disparities in subtitling tradition and conventions.

The biggest multi-language subtitles database is www.opensubtitles.org[11]. The entries consist of user uploads and cover 18,900 movies in 59 languages. Every subtitle is associated to a time code in the same movie or TV episode, saved as a metadata, which makes it possible to align them efficiently across different languages or within the same language to obtain alternative translations, because corresponding segments are shown approximately at the same time.

To create a corpus from this database, it was necessary to clean and filter the original sources, to avoid incorrect language or titles tags added by the users and make character encoding and format uniform. The present work is based upon the latest available version, built in 2018.

As Tiedemann (2007) describes, first they used a language encoding table to convert or remove characters not belonging to UTF-8 in a reliable, even if not complete way. Then, with **textcat**[12], a language classifier trained on N-gram models, they checked if the language of the document matched the tag;

[11] http://www.opensubtitles.org
[12] https://www.let.rug.nl/~vannoord/TextCat

removed HTML tags present in the text and used regular expressions to tokenize and split the sentences. Finally, only subtitle pairs made for the exact same movie file were chosen.

After the cleaning process, they selected approximately 3.73 million subtitles in 60 languages to include in OPUS.

The corpus uses the XCES (XML Corpus Encoding Standard) format, one of the most common for parallel/multilingual corpora. The respective monolingual corpora are stored separately and there is one or more documents for the alignment. This makes it easier to create sub-corpora, change the alignment, and go back to the original source. Each subtitle file is saved in XML format with an ID, the path to the file or list of files containing the text entries, the language code, information about the source material and other file attributes.

Other than at the document level, the corpus is aligned at the sentence level using a variation of the traditional length-based approach introduced by Gale and Church (1993) that uses time lengths instead of the number of characters (**linear time algorithm**, Tiedemann, 2008). This is necessary because, being the translations less literal than in other text types, the insertions and deletions would otherwise cause too many follow-up errors. Along with the matching sentences, a confidence score of the alignment is also saved, calculated with the following heuristics: the sentences must include at least some matching tokens; the string in one language cannot be more than twice as long as the other, unless the time overlap is over 90%; the alignment is more likely to be wrong if it follows a mis-aligned pair.

The alignment at the token level is calculated with models trained running **GIZA++** (Och and Ney, 2003) on the entire parallel data set and a symmetrization heuristic implemented in **Moses** (Koehn et al., 2007) to extract probabilistic phrase tables. OPUS also offers a tool to explore these models through a search interface, but the word alignment tags are currently not available in the corpus, even if there are plans to include them in future versions (Tiedemann, 2012).

The subtitles corpus represents a unique resource in terms of size and language variety and its features make it an ideal candidate for the purposes of this work. A possible issue is the reliability of the data. As was said, the subtitles are submitted voluntarily by users, with a **crowdsourcing** method so the translations are not necessarily made by experts. If the same users produce more data, they probably can be trusted, and it is always possible for other

people to update the subtitles or to report errors, but the quality is not guaranteed as the providers of the subtitles translation platform cannot check every document that is uploaded. Despite this, the corpus has already been successfully used by many NLP applications, like Reverso[13], a translator of sentences in context, or Sketch Engine[14], a corpus management system with concordancing and text analysis functionalities for linguists and lexicographers. To the author's knowledge, this is a first attempt at creating a tool for language learning with it.

The sentences are available either in plain-text, or annotated with parsing information, if a state-of-the-art pipeline is available for the language. This project uses the parsed versions of the English, Italian and Swedish corpora. The tags available in OPUS for the Swedish and English documents were processed with **HunPos** (Halácsy et al., 2007) and **MaltParser** (Nivre et al., 2007), while the Italian ones with **TextPRO** (Pianta et al., 2008) and MaltParser.

## 3.2 WORD ALIGNMENT

Word alignment was first introduced as a supporting task for statistical machine translation (MT). The idea, "both algorithmically appealing and empirically successful" (DeNero and Klein, 2007), was to find correspondences between words or multi-words units to factor the translation model (Brown et al., 1993). In Galley et al. (2006) it was applied to project a tree from target language to source, in Chiang (2005) to induce grammar rules for MT. More recently, word alignment has been used for lexicon injection or annotations transfer because statistical MT has been replaced by neural approaches, making this step unnecessary.

As was described in the previous section, word alignment can also be applied to corpus development and sentence alignment evaluation (Zariņa et al., 2015), which often use the same metrics because the quality of a corpus is affected by the number of erroneously aligned sentences. If two sentences are aligned correctly and are translations of each other, they are expected to have a higher number of alignments at the word level compared to non-parallel texts, so it is

[13] https://context.reverso.net/translation
[14] https://www.sketchengine.eu

common to clean a corpus by filtering sentence pairs with a low number of aligned words.

The most widely used tools for word alignment are GIZA++, the standard pipeline introduced by Brown et al. (1993), which still performs competitively but it uses a lot of resources (CPU time, RAM, etc.); and **FAST_ALIGN** (Dyer et al., 2013), an efficient unsupervised procedure based on parameter estimation.

Some attempts have been done with neural methods. Zenkel et al. (2020) describe a pre-trained neural translation model with an alignment layer that outperforms GIZA++ in terms of error rate.

Other approaches extend GIZA++ in some way. The **Berkeley aligner** (DeNero and Klein, 2007) is based on the intuition that alignment vectors are locally monotonic, with few larger jumps. It is also an unsupervised method and it uses HMM models, with an alignment vector that specifies the position of an aligned target word for each source word.

The present work uses **EFLOMAL**[15], an easy, unsupervised, low-memory alignment tool which estimates the alignments one sentence at a time and has proven to be accurate and computationally efficient. There are no direct comparisons in research between EFLOMAL and the other methods, Östling and Tiedemann (2016) tested its predecessor **EFMARAL**[16] against FAST_ALIGN and GIZA++, obtaining better results especially in terms of processing time. It is beyond the scope of this work to analyse if the improvements achieved by this method are higher or lower than the ones obtained with the neural approach.

Like EFMARAL, GIZA++, and FAST_ALIGN, EFLOMAL builds on the **IBM models** (Brown et al., 1993), five increasingly complex statistical models built to describe how a source language generates a target language through a set of alignment variables.

An alignment is not necessarily 1:1 between two tokens, it can also be 1:n, even if n is generally low in a natural language. For this reason, the alignment is defined between *cepts*, a term introduced to indicate words or multi-words units which represent the same concepts, e.g. *implemented* in English

[15] https://github.com/robertostling/eflomal
[16] https://github.com/robertostling/efmaral

corresponds to *mis en application* in French, and a word may participate in more than one *cept*.

The first IBM models, 1 and 2, are very low level, they only consider the length of the strings and the order of the sentence. They are often insufficient to find satisfactory correlations, but they are the only ones in which it is feasible to examine all the possible alignments. Model 3 adds the concept of **fertility** of a source word: the number of words in the target language it will be connected to. Model 4 and model 5 include the position of other strings connected to the same word in distinct sentences (**identity** of the word), with the difference that model 4 "wastes" some of its probability on objects that are not words of the language. To guarantee that all the possibly meaningful alignments are explored, every model is initialized from the parameters estimated by the one trained before it.

Most researchers who used IBM alignment models trained them iteratively with maximum-likelihood estimation through Expectation Maximization (EM). This approach does not consider the sparsity of the lexical distribution in natural language, because there are no constraints preventing one word to have a large number of target words as possible translations. Östling and Tiedemann (2016) incorporate sparse and symmetric Dirichlet priors to add these constraints and proved they are valuable to find more realistic solutions. Instead of EM, which does not allow priors, they used Gibbs sampling, a special case of the Markov Chain Monte Carlo (MCMC) method.

It is essential to have reasonable priors of the parameters to obtain high levels of accuracy. This is not always possible, especially when the document to align is small, being the method unsupervised and only based on statistics. A solution is to use a large file as training data and store the generated priors. These can then be used to initialize the parameters and align even very small documents with comparable accuracy.

## 3.3 COMPLEXITY OF A SENTENCE

Unlike parsing and alignment, assessing the complexity of a sentence does not have a standardized solution. As described in section 2.1.2, there are several possible approaches and the best solution may not be the same for every language and application.

At the state-of-the-art, there are ready-to-use tools available only for some languages. This work examines the approach used by **Tint** (Aprosio, Moretti, 2018), and **HitEx** (Pilán et al., 2016).

Tint is an open-source NLP suite built on **Stanford CoreNLP** (Manning et al., 2014) meant to organize in a single framework most of the standard NLP modules, from tokenizer to part-of-speech (POS) tagger, morphological analyser and parser, and optimize the pipeline for the Italian language.

The readability module is inspired by two earlier works: READ-IT, which cannot be used on its own because it is available only in the form of an online demo; and Tonelli et al. (2012), a system based on Coh-Metrix to calculate the readability of a document at three levels of proficiency: elementary, middle and high-school. The metric is calculated from a series of indices:

- Number of content words, hyphens, and distribution of tokens based on POS
- Type-token ratio (TTR) between the number of different lemmas and the number of tokens
- Lexical density, number of content words divided by the total number of words
- Amount of coordinate and subordinate clauses, along with the ratio between them
- Average and max depth of the parse tree
- Gulpease formula[17] (Lucisano and Piemontese, 1988) to measure the readability at document level
- Text difficulty based on word lists from DeMauro's Dictionary of Basic Italian[18].

The Tint pipeline is released under the GNU General Public Licence, version 3 and it is written following the Standford CoreNLP paradigm, so it is possible to integrate it into an application and to extend or replace its modules.

The current version of the readability module supports four languages: English, Spanish, Galician, and Italian. For the languages other than Italian, the pipeline

[17] Index created specifically for the Italian language, based on the number of characters contained in each word. It is calculated as follows:

$$89 + \frac{300 * (number\ of\ sentences) - 10 * (number\ of\ letters)}{number\ of\ words}$$

[18] http://bit.ly/nuovo-demauro

leading to the computation of the metric is the one from the Standard CoreNLP library, included in Tint.

HitEx (*Hitta Exempel*, "Find Examples") is a hybrid system using a combination of machine learning methods and heuristic rules. The supervised machine learning approach exploits available data to assess the complexity of sentences, considering multiple linguistic dimensions at the same time, while the rules make the selection customizable to task-specific needs (see section 2.1.2).

| Nr | Criterion | Nr | Criterion |
|---|---|---|---|
| | **Search term** | | **Additional structural criteria** |
| 1 | *Absence of search term* | 13 | Negative formulations |
| 2 | Number of matches | 14 | *Interrogative sentence* |
| 3 | *Position of search term* | 15 | *Direct speech* |
| | **Well-formedness** | 16 | *Answer to closed questions* |
| 4 | *Dependency root* | 17 | Modal verbs |
| 5 | Ellipsis | 18 | Sentence length |
| 6 | *Incompleteness* | | **Additional lexical criteria** |
| 7 | Non-lemmatized tokens | 19 | Difficult vocabulary |
| 8 | Non-alphabetical tokens | 20 | Word frequency |
| | **Context independence** | 21 | Out-of-vocabulary words |
| 9 | *Structural connective in isolation* | 22 | Sensitive vocabulary |
| 10 | Pronominal anaphora | 23 | Typicality |
| 11 | Adverbial anaphora | 24 | Proper names |
| 12 | **L2 complexity in CEFR level** | 25 | Abbreviations |

*FIGURE 2 – HitEx Criteria (table from Pilán et al., 2016)*

**Figure 2** shows the criteria associated to the sentence goodness. Some of them are used as filters, as they target negative aspects like incompleteness, non-alphabetical tokens, and anaphora, others as rankers to compute a **goodness** score. Other than checking for the presence or absence of linguistic elements, to make the system more pedagogically aware the latest version of HitEx filters out sensitive vocabulary and considers the frequency of words from **SVALex** (part of the CEFRLex resources; François et al., 2016), a list based on coursebooks text, and the presence of lemmas associated to a higher proficiency level in the **KELLY** list (Volodina and Kokkinakis, 2012).

Second language (L2) complexity is calculated on a 5-level scale using a supervised machine learning classifier trained on **COCTAILL** (Volodina et al., 2014), a corpus of coursebook texts for L2 Swedish containing sentences chosen to exemplify a lexical or grammatical pattern. For each of these sentences, features were considered from five categories: count-based, for example the number of characters or tokens; lexical; morphological; syntactic; and semantic. Pilán et al. (2016) evaluated this classification method obtaining an accuracy of 63.4% for exact matches and 92% within one CEFR level.

The sentence selection algorithm is integrated into the learning platform **Lärka**[19]. It can be accessed through the online graphical interface or as a web service.

## 3.4 EXERCISE

The tools described until now are used to select and process sentences from the OpenSubtitles parallel corpus with the purpose of generating language learning exercises. The same pipeline can be used for several types of exercises, in this work we exemplify its potential through one type.

Since it is difficult to ensure an accurate feedback without any human intervention, most of the automatically generated exercises in ICALL literature are fill-in-the-blanks or multiple-choice questions, which have very limited answer options. In the present work, the proposed type is **sentence reconstruction**.

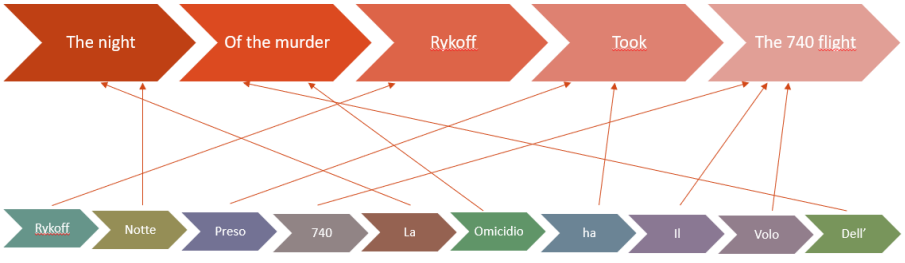



*FIGURE 3 – Sentence reconstruction exercise for Italian learners with English as source language.*

[19] https://spraakbanken.gu.se/larkalabb/hitex

Each sentence is split into tokens (or units of meaning, as explained later), which are given to the user in a random order along with the sentence's translation in their native language. The learner is then asked to re-order the sentence, finding matching units between the source and target sentence (**FIGURE 3**).

It combines two elements which have proven to be effective for language learning: (1) a game-like approach and (2) the identification of syntactic structures and vocabulary use.

Games are used by teachers for language learning at every level and age group. They create an environment where education is mostly learner-centred and can be designed to teach a specific skill. In recent years, experiments have been performed to see if digital games could increase language acquisition (Klimova, 2017). The results showed that Game-Based Learning (GBL) awakens competitive desire, making the learner more engaged and motivated. If the game is not too complex, in which case it distracts from the language used, it encourages to observe the data and identify patterns, and improves vocabulary acquisition.

The recognition of syntactic structures through comparing a familiar expression to an equivalent one in the new language, favours the users' language sensitivity, making them more autonomous. Students can look at the translation to confirm the specific meaning of an expression in the target language and formulate their own ideas about its use. Since human cognition is based on pattern detection, a rule which learners come up with themselves will be more meaningful and relevant to them (Cobb, 1999).

Instead of using each word as a token, the system we implemented separates larger units of meaning. The hypothesis is that larger units are more relevant for language learners, because a word without its context does not have a definite meaning, while clusters of words highlight syntactic structures and concepts. This is especially true in the case of function words, which often just perform a syntactic function and they are difficult to align without the association to a content word. In the example in **FIGURE 3**, it would not be necessarily correct to say that the word *dell'* in Italian is the equivalent of *of the* in English, in a different context, the meaning of the preposition *dell'* may be expressed by another word, but the expression *dell'omicidio* can always be translated as *of the murder*. Focusing on multi-word units instead of words, then, increases the accuracy of the translation. According to Boulton (2017),

psycholinguistic works on **chunking** support the idea that the mind works with exemplars beyond the level of words. They suggested to choose clusters/n-grams with the same number of words in each string, to see how they group together, even if these clusters may not carry much meaning.

Other works based on clusters of words can be found in Byrd and Coxhead (2010), where four-word bundles are extracted from a corpus to help learners improve their academic writing, noting specific formulations.

Cobb and Boulton (2015) ran a three years experiment and demonstrated the words met through concordances are retained in the 75.9% of the cases, against the words met through simple definitions, which are retained only in the 63.9%.

Wu et al. (2014) introduced the idea of **lexical bundles**, multi-words units with distinctive syntactic patterns and discourse functions to identify the ones most common in academic prose.

# 4 IMPLEMENTATION

The author's main contribution is the definition and implementation of the pipeline shown in **FIGURE 4** and the manual evaluation of the results.

At the end of the processing steps (step 4), the result is an object containing the text of the sentence in two languages and the following information:

- POS and parsing tags, obtained from OPUS along with the text.
- Word alignment, processed with EFLOMAL.
- *Word clusters*.
- Proficiency level, calculated with Tint for Italian and English and HitEx for Swedish.

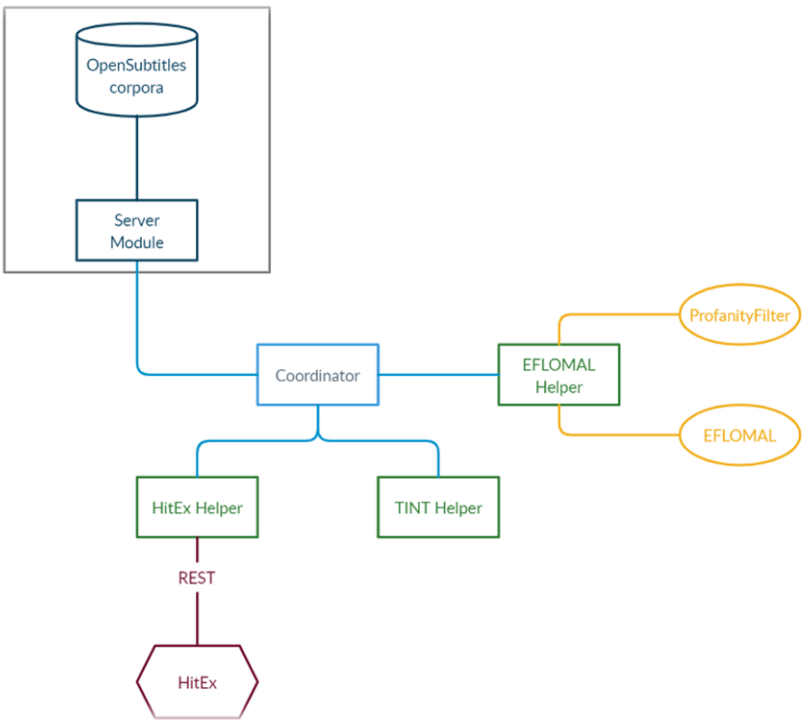



*FIGURE 4 – Components Diagram*

The project has been developed in three different components (**FIGURE 4**):

- Java project: acting as coordinator. It is the main access point to the application; it runs all the components for the specified language pair in sequence and produces a JSON file with the results. Other than coordinating the modules, it contains the Tint and Stanford Core NLP libraries, used to compute the proficiency level of the Italian and English sentences, and the **REST** architecture designated to communicate with HitEx.

- Server module: a Python module meant to run on the server containing the corpora. It has two main functions, the first one selects the candidate sentence pairs from the OpenSubtitles corpus, the second one filters them according to the alignment at the word level and extracts the word clusters, necessary for the proposed exercise. The connection to the server is handled using the Python library **paramiko**[20], which implements the SSH protocol, both to run the module and to upload or download the necessary files.

- Client module: a Python module that runs on the client ("EFLOMAL Helper" in the figure). It filters the inappropriate sentences and prepares the data for the word alignment step, which is delegated to the EFLOMAL Python library.

A configuration file allows the deployment of the project on a different environment. When the application is run for the first time, the server module is copied on the specified server, making the application portable.

Each module is also available on its own if the user is interested in running only one part of the project.

## 4.1 SENTENCES SELECTION

The candidate sentences are proposed by the *setup* function. First, the module reads the sentence alignment file for a language pair and randomly selects from

[20] http://www.paramiko.org

the parallel corpus the desired number of documents. A utility function saves into a separate file the ids of the documents already processed in previous iterations, to prevent the system from choosing the same document more than once. Then, the sentence pairs in the documents are extracted and discarded if the overlap is lower than 50% or the match is not 1:1.

The module can be run without a specified number of documents. In that case, all the documents in the corpus are processed, obtaining 31,366,775 sentence pairs for Italian-English, 6,471,759 for Swedish-English, and 3,828,688 for Swedish-Italian.

The output of this step is a .txt file that can be used to run EFLOMAL, in the format:

*Sentence pair ID* ||| *Sentence 1* ||| *Sentence 2*

The *sentence pair ID* is the key associated to the alignment in the OpenSubtitles corpus, followed by the text of the two sentences. The file is returned to the client application to proceed with the word alignment.

The EFLOMAL Helper module checks that every line in the file is in the correct format and that the sentences are not empty, then it processes them with a **profanity filter**[21], to exclude examples inappropriate for a language learning environment.

The profanity-filter Python library extends an English profane words dictionary extracted from Google[22] to support text in mixed languages and derivative and distorted words. Each word is considered without its context and labelled as profane or safe. For this reason, a safe word used in a profane sentence or vice versa will be labelled incorrectly.

A more accurate filter is the one used in HitEx, currently available for Swedish, which excludes sensitive vocabulary items but also topics that tend to be avoided in a pedagogical setting, referred to as **PARSNIP** (Politics, Alcohol, Religion, Sex, Narcotics, Isms - such as communism or atheism, and Pork; Gray, 2010).

[21] https://pypi.org/project/profanity-filter
[22] https://github.com/areebbeigh/profanityfilter

The HitEx filter contains a list of 263 items and will be expanded in the future (Pilán et al., 2017). It was created starting from a group of seed words from undesirable domains, collected from online resources like Wikipedia and integrated manually, and expanding them with all the child node senses from **SALDO** (Borin et al., 2013) for terms which represent sensitive topics, to include synonyms and hyperonyms.

With the examples found in the OpenSubtitles corpus, the profanity-filter library has proven to give acceptable results for the English and Italian languages, and it is both easy to integrate and efficient. Since HitEx is an external web-service and it is called in the last step to calculate their proficiency level, the Swedish sentences are not processed in this phase. The information about the eventual inappropriateness is obtained along with the CEFR level and an additional filter is realized accordingly.

The remaining sentences are aligned with EFLOMAL. To make sure the alignment is accurate even when processing one document at a time, the module is run with previously generated priors in both directions, forward and backward, which are symmetrized using a function integrated in EFLOMAL.

To generate the priors, the setup function has been used on a large number of documents to obtain approximately 10,000,000 aligned sentences for language pair. The values are saved on a file in the client, whose name can also be specified in the configuration file.

The output of EFLOMAL is a file in the format:

*0-0 1-1 2-2 5-3 6-4 7-5 8-6 9-7 10-8 12-9 13-11*

A number is associated progressively to each token in the source and target sentences and the matching tokens are aligned using the "-" symbol.

For example, the alignment showed before corresponds to the following sentences:

English:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Every | day | when | one | 's | body | and | mind | are | at | peace | one | should | meditate |

Swedish:

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Varje | dag | när | kropp | och | själ | är | i | ro | skall | man | meditera |

It is read as *Every-Varje*, *day-dag*, *when-när*, *body-kropp*, *and-och*, *mind-själ*, *are-är*, et cetera.

The results are three files, one containing the cleaned sentences, the other containing the forward and backward word alignments.

Since the goal is to use the parallel sentence in a language the user understands to solve the exercise, the translations need to be as close to literal as possible. Following the intuition from Bourdaillet et al. (2010), the sentence alignments are considered "bad" if the POS of their tokens do not match.

The sentences, along with the word alignment files, are processed by the second and key function in the server module, which chooses the good matches and separates the sentences into word clusters.

Sentence pairs are excluded if either of the sentences has less than 5 tokens or does not have a finite verb, because it is likely to be a partial or elliptic sentence, and thus dependent on the context. Then, the POS of each word in the source language is compared to the POS of the aligned word in the target language to check if they are the same. To increase the tolerance, the non-matching POS tags are excluded if they are **auxiliary verbs** (AUX), **articles** (DET), or **punctuation marks** (PUNCT) in either of the sentences, as the first tests showed these elements do not affect the accuracy of the selection and many promising sentences would be excluded.

In the following examples the non-matching tokens are shown in bold:

*1* *(EN)* What color am **I** thinking of
*(IT)* Che colore **sto** pensando

*What (DET) – Che (DET)*
*color (NOUN) – colore (NOUN)*
*am (AUX) –*
***I (PRON) – sto (AUX)***
*thinking (VERB) – pensando (VERB)*
*of (ADP) –*

| | | |
|---|---|---|
| *2* | *(EN) A* samurai must always **stay** loyal to **his** boss<br>*(SW)* En samuraj måste alltid **vara** lojal mot **sin** herre | *A (DET) – En (DET)*<br>*samurai (NOUN)* – samuraj (NOUN)<br>*must (AUX) – måste (AUX)*<br>*always (ADV) – alltid (ADV)*<br>***stay (VERB) – vara (AUX)***<br>*loyal (ADJ) – lojal (ADJ)*<br>*to (ADP) – mot (ADP)*<br>***his (PRON) – sin (DET)***<br>*boss (NOUN) – herre (NOUN)* |
| *3* | *(IT)* Aspetto la **vostra** risposta<br>*(SW)* Jag väntar **på** ert svar | *Aspetto (VERB) – väntar (VERB)*<br>*– Jag (PRON)*<br>***la (DET) – på (ADP)***<br>*vostra (DET) – ert (DET)*<br>*risposta (NOUN) – svar (NOUN)* |
| *4* | *(EN)* **-** You read that<br>*(SW)* **Har** du läst den | ***- (PUNCT) – Har (AUX)***<br>*You (PRON) – du (PRON)*<br>*Read (VERB – last (VERB)*<br>*That (PRON) –*<br>*– den (PRON)* |

*TABLE 1 - Examples of included sentences with non-matching POS*

The selection procedure was tested manually, tagging the results obtained processing 10 random documents from the corpus for each language pair. The Italian-English documents contained 5575 sentence pairs; 430 were selected and 392 were considered appropriate for the purposes of this work because they were both good translations and usable for an exercise without context, obtaining a precision of 91.16%.

The 38 "bad" sentences were excluded for several reasons, the following table shows the main categories that were found.

| | | |
|---|---|---|
| *Missing part* | *(EN)* I know that guy | *(IT)* Quello lo conosco **è un poliziotto** |

| | | |
|---|---|---|
| *Non-literal translation* | *(EN)* You **rolled away the stone** | *(IT)* Sei **uscito dal sepolcro** |
| *Missing context* | *(EN)* My lawyer says that | *(IT)* Il mio avvocato ha |
| *Figure of speech* | *(EN)* I need **a bird in the air** right now | *(IT)* Mi serve **un elicottero** subito |

*TABLE 2 – "Bad" sentences English - Italian*

For the English-Swedish pairs the results were not as good, especially because the Swedish sentences were, in the majority of the cases, summary of the English sentence instead of literal translations. Some examples can be accepted, because the additional token is one that does not influence the meaning, like "well," or "yes", but many pairs had to still be excluded because the syntactic structure of the sentences differed to much to be used for language learning purposes. In the examined documents, many sentences were also opposite in terms of negative wording/positive wording or active form/passive form between the two languages.

The algorithm selected 598 out of the 5571 pairs present in the documents, with an estimated precision of 72.74%.

Not all the pairs were tagged as "bad" because they were not matching, some were also excluded because the profanity filter did not work for Swedish, so these examples would presumably be excluded at a later stage by HitEx.

| | | |
|---|---|---|
| *Missing part* | *(EN)* **Kind of, sort of** want to emulate you, **you being** my mentor **and all** | *(SW)* Jag vill väl efterlikna dig, min mentor |
| *Non-literal translation* | *(EN)* No, **we lost him** | *(SW)* Nej, **han är död** |
| *Missing context* | *(EN)* But **the machine** said this was Carol | *(SW)* Men **den** sa att det var Carol |
| *Opposite wording* | *(EN)* **You have to give me** an injection | *(SW)* **Jag måste få** en spruta |

| | | |
|---|---|---|
| *Positive/Negative* | *(EN)* Other mums **wouldn't** | *(SW)* Annan mamma **skulle** göra så |
| *Different measures* | *(EN)* It's **200 miles** away | *(SW)* Den är **300 km** härifrån |

*TABLE 3 – "Bad" sentences English - Swedish*

The situation is similar for the Italian-Swedish pairs, some of the sentences used in the test were selected because their structures looked similar, even if the meaning was different. For example, the Italian sentence:

| La | tua | **proposta** | **è** | una | **follia** |
|---|---|---|---|---|---|
| DET | DET | **NOUN** | **AUX** | DET | **NOUN** |

Was aligned with the Swedish, even if the meaning is completely different:

| **Tiden** | **är** | knapp | **broder** |
|---|---|---|---|
| **NOUN** | **AUX** | ADJ | **NOUN** |

Some were also tagged negatively for appropriateness reasons.

| | | |
|---|---|---|
| *Missing part* | *(IT)* **Questa è la mia diocesi,** quindi condurrò io il processo | *(SW)* Det är min plikt att genomföra rättegången |
| *Non-literal translation* | *(IT)* L' onestà non fa **parte del gioco** | *(SW)* Ärlighet har ingen **betydelse** |
| *Missing context* | *(IT)* Se indosserai **quell'** uniforme, non potrai indossare anche la mia | *(SW)* Om du ska bära **polisens** uniform får du inte ha min |
| *Opposite wording* | *(IT)* **Sono stata minacciata** | *(SW)* **De hotade mig** |

*TABLE 4 – "Bad" sentences Italian-Swedish*

The documents in total contained 5124 pairs and the algorithm selected 306 of them, with an estimated precision of 69.28%.

## 4.2 WORD CLUSTERS

The intuition that clusters of words can be useful for language learners has been considered in several works with different approaches, as was shown in the previous chapter. Yet, it still has to be proven from a pedagogical point of view.

In some cases, the idea of a cluster makes it possible to find correspondences between an expression in the target language and one in the source language.

The following are sentences from the OpenSubtitles corpus:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| *(EN)* | We | ‘ve | **turned** | this | place | **upside** | **down** |
| *(IT)* | | Abbiamo | **ribaltato** | questo | posto | | |

The concept of the Italian verb *ribaltato* is not captured by the English *turned* alone, but it is a perfect translation of *turned upside down*.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| *(EN)* | I | **‘ve** | **been** | **looking** | everywhere | for | you |
| *(SW)* | Jag | **har** | | **letat** | överallt | efter | dig |

The English auxiliary *been* does not match directly with any of the Swedish tokens, although the form *have been looking* can be translated with *har letat*.

| | | | | | | |
|---|---|---|---|---|---|---|
| *(IT)* | Suo | **marito** | era | **un** | **uomo** | **crudele** |
| *(SW)* | Din | **make** | var | | | **ond** |

The Italian sentence is more verbose, because the concept of *marito* (husband) implies the concept of *uomo* (man), so the meaning of the sentence would be

the same if it were *Suo marito era crudele*, which matches the Swedish wording exactly. In this case, to say *un uomo crudele* expresses the same meaning as the word *ond*.

From the list of the Universal Dependency tags[23], the possible POS and DEPREL values were separated into **core** and **dependent**. Each cluster has one core element and all its dependents.

The subdivision was done following two rules:

1. A token is identified as core if its POS tag is "NOUN", "PROPN" or "VERB", unless its dependency relation is "compound", "name", "mwe", "goeswith", "aux", "auxpass", "case"; or its POS tag is "CCONJ", "SCONJ" or "INTJ".

2. A token is identified as core if its dependency relation is "nsubj", "nsubjpass", "csubj", "csubjpass", "ccomp", "xcomp", "obj", "iobj", "obl".

To identify the groups of tokens, we used a **divisive hierarchical clustering** strategy: a top-down approach which starts by including all the objects in one single cluster, then processes the nodes iteratively to decide where to separate the elements, adding a smaller cluster.

In the example in **FIGURE 5**, the sentence *She knows that we go all the way back to the academy* was parsed starting from the node *knows* (POS=VERB, DEPREL=ROOT).

The algorithm examines the branches iteratively, until all the nodes have been explored:

- The first one is *she* (POS=PRON, DEPREL=NSUBJ). For the first rule, it would be included in the same cluster, because its POS tag is "PRON", but the relation is "NSUBJ", so for the second rule a new cluster is created.
- The second node is *go* (POS=VERB, DEPREL=CCOMP). It is excluded for the first rule, because its POS tag is "VERB".
- Etc.

[23] https://universaldependencies.org

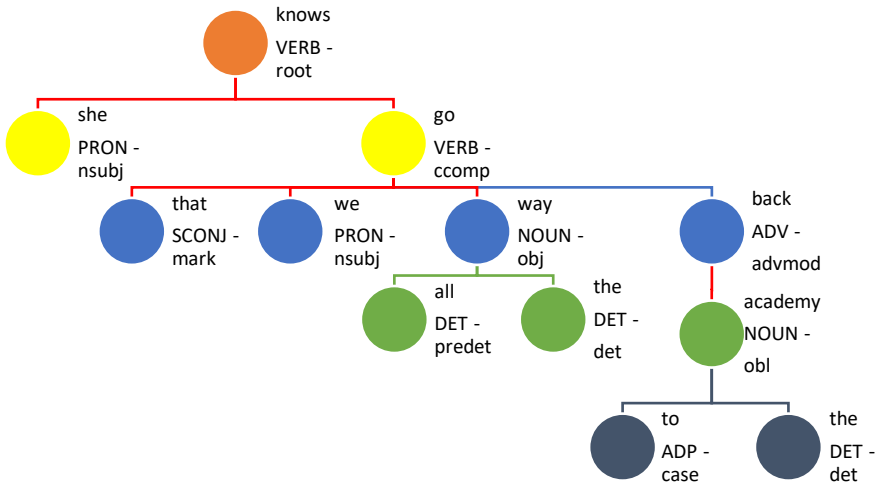



*FIGURE 5 – Parsing tree*

The clusters obtained in the end are 7: [ *she* ], [ *knows* ], [ *that* ], [ *we* ], [ *go back* ], [ *all the way* ], [ *to the academy* ]. If the branches have been cut, they are colored in red in the figure.

The tree is represented using a custom object **Node**, which contains the information about the token (*id*, *aligned_id*, *head*, *deprel*, *pos*, *word*, *lemma*), and a link to its dependents in the parsing tree.

The results are then saved in the format:

*{ token1 } || { token2 } { token3 } ||*

A token is the serialization of one of the Node objects and the "||" symbol separates them into the clusters.

There is not always only one possible solution, for example, in the previous sentence, it could be argued that *back* should go in the same cluster as *all the way*. In other cases, the parse tree is not perfect, causing inaccurate clusters. TextPro has been evaluated at EVALITA 2007 obtaining an accuracy of 98.04% for Italian (Pianta et al., 2008), while the accuracy of MaltParser has been estimated as 85% for Swedish and 91% for English (Nivre, 2003).

The algorithm was tested on the same 30 documents used for the candidate sentence generation and, even if the author agrees with the annotation in the majority of the cases, some sentences were found in which small inaccuracies in the parsing trees caused "wrong" separations (**TABLE 5**). In order to properly evaluate this phase, a pedagogical perspective on the clusters would be needed.

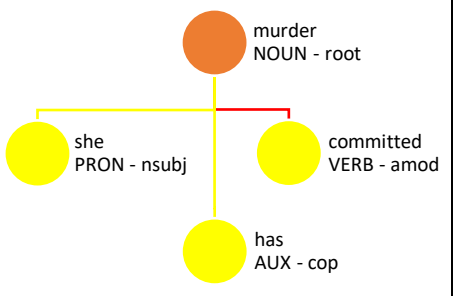
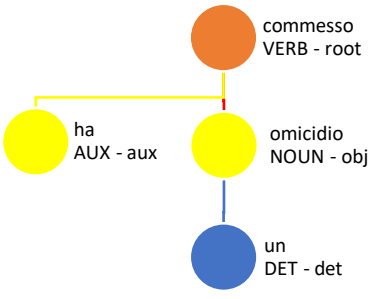
| *[EN]* She has committed murder | *[IT]* Ha commesso un omicidio |
|---|---|
|  |  |
| She has *murder* \|\| **committed** | Ha **commesso** \|\| un *omicidio* |

*TABLE 5 – Example of "wrong" clusters*

Other than to create a different type of exercise, the clusters can be used to improve the precision of the match between the sentences in the source and target language. As was showed in the previous section, many mistakes were due to one of the two sentences being more verbose than the other, but it would be too restrictive to only keep the pairs with a 1:1 match between the tokens. Different languages can express the tense of the verbs with more or less tokens or a different meaning with a particle instead of a new word.

Using the tagged documents from the previous section, two experiments were run to exclude all the sentence pairs in which there is at least one cluster without any match in the other language.

In the first one only auxiliaries and determiners were tolerated even if they did not have a match because they do not affect the meaning, like in the sentence selection.

The extracted sentences are fewer, but the accuracy increases significantly, especially for the English-Swedish and Italian-Swedish pairs:

- English – Italian: 154 sentences (37.76% recall) with an estimated accuracy of 96.10%
- English – Swedish: 316 sentences (65.98% recall) with an estimated accuracy of 90.82%
- Italian – Swedish: 97 sentences (38.21% recall) with an estimated accuracy of 83.51%.

In the second experiment, the clusters were included even if the token without a match was a pronoun, because it was noticed, especially in the pairs including Italian, that the meaning of the pronoun was often included in the verb through inflection, while it was explicit in English and Swedish.

The results are slightly worse in terms of precision even if still acceptable and the recall improves significantly:

- English – Italian: 308 sentences (75.51% recall) with an estimated accuracy of 96.10%
- English – Swedish: 394 sentences (79.31% recall) with an estimated accuracy of 87.56%
- Italian – Swedish: 200 sentences (76.89% recall) with an estimated accuracy of 81.50%.

## 4.3 PROFICIENCY LEVEL

The final step is the estimation of the complexity of the sentence. As was shown in **FIGURE 4**, this is done by two separate components: the TINT Helper and the HitEx Helper.

The first includes the Java Stanford Core NLP and Tint libraries and it is used for the Italian and English sentences. The second realizes a REST architecture to call the web service HitEx, available for Swedish.

The Italian pipeline is the default Tint pipeline. It pre-processes the raw text of the sentence and runs the **readability module** to produce a numerical value that represents the main calculated complexity of the sentence. A higher measure means the proficiency level needed to understand the text is lower.

For English, the component uses the same Tint module to compute the readability measure, but it relies on the standard Stanford Core NLP modules to pre-process the raw sentence, because the Tint pipeline would not have the same results, being optimized only for Italian.

Completely different is the Swedish component. The HitEx web service is available through **KORP**, Språkbanken's corpus query infrastructure[24]. It can be accessed using an URL in the following format, with the desired parameters after the "&":

*https://ws.spraakbanken.gu.se/ws/larkalabb/icall.cgi?command=hitex&*

It returns a JSON object containing the eventual match and its calculated CEFR level between A1 and C1.

In order to access the documents from OpenSubtitles, it was necessary to pre-emptively integrate them into KORP, creating a sub-section called *OPUS-OPENSUBTITLES-SV*.

HitEx was created to extract candidate sentence given a target word or lemma, but there is also the possibility to use the Corpus Query Language CQP[25]. Since in this work, the interest is in evaluating a specific sentence, the CQP modality was used, to pass the entire sequence of tokens as a parameter.

| | |
|---|---|
| **query_type** | *cqp* |
| **query_w** | " sentence " |
| **corpus_list** | *OPUS-OPENSUBTITLES-SV* |
| **max_kwics** | *100* |
| **maxhit** | *20* |
| **target_cefr** | **C1* |
| **readability** | *filter* |
| **preserve_bad** | **true*[26] |
| **random_seed** | *2* |

*TABLE 6 – List of parameters used to call HitEx and their values*

[24] https://spraakbanken.gu.se
[25] http://cwb.sourceforge.net
[26] *It is necessary because the web-service requires a CEFR level as a parameter. In the query the value is always "C1", but the service returns the matching sentence even if the level is different thanks to this parameter

If there is no match, the sentence is excluded from the selection, otherwise the system extracts the calculated CEFR level.

The numerical value calculated by the Tint Helper can be separated using thresholds: a sentence with a readability over 80 is understandable by students from an elementary school, over 60 by students from a middle school and over 40 by students from high school.

For the purpose of this work, it was necessary to obtain a complexity measure to separate the sentences into easy, medium or difficult. Considering in general, the elementary school level is estimated as A1, middle school as A2 and high-school as B1, the sentences were divided into 4 categories, as showed in **TABLE 7**.

| Level | Italian - English | Swedish |
|---|---|---|
| 0 | Sentence with readability over 80 | Sentence with CEFR level "A1" |
| 1 | Sentence with readability over 60 | Sentence with CEFR level "A2" |
| 2 | Sentence with readability over 40 | Sentence with CEFR level "B1" |
| 3 | Other | Other |

*TABLE 7 – Estimated complexity*

The components were tested on the sentences selected after the experiment from section **4.1** and annotated as correct.

Out of 392 Italian-English sentence pairs, 8 could not be calculated, the others were estimated as showed in **TABLE 8**.

| English ↓ / Italian → | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 252 | 32 | 0 | 0 |
| 1 | 60 | 15 | 0 | 0 |
| 2 | 11 | 5 | 1 | 0 |
| 3 | 7 | 1 | 0 | 0 |

*TABLE 8 – Detail of the complexity in the Italian-English sentences*

The English-Swedish pairs were 435 and 5 could not be calculated.

| **English ↓ / Swedish →** | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 251 | 70 | 0 | 4 |
| 1 | 54 | 27 | 0 | 6 |
| 2 | 9 | 5 | 0 | 0 |
| 3 | 4 | 0 | 0 | 0 |

*TABLE 9 – Detail of the complexity in the English-Swedish sentences*

The Italian-Swedish pairs were 212 with no errors.

| **Italian ↓ / Swedish →** | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | 148 | 45 | 0 | 4 |
| 1 | 7 | 7 | 0 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |

*TABLE 10 – Detail of the complexity in the Italian-Swedish sentences*

# 5 CONCLUSIONS

The research question of this work was how natural language processing can be applied on a parallel corpus for second language learning. To expand the problem, it was separated into two sub-parts:

1. Is it possible to extract candidate sentences from a parallel corpus to generate exercises for second language learning without human intervention?

The answer is yes, it is possible. The sentences included in the OpenSubtitles corpus are in the majority of the cases short, analysing 30 random documents, the system found 2024 “good” sentences but 690 of them were discarded because they had fewer than 5 tokens. The size of the corpus allows to favour precision over recall, using very strict criteria to select candidate sentences for language learning exercises without any human intervention, obtaining a large number of examples with acceptable levels of accuracy.

2. Which pre-processing operations are necessary?

The pipeline presented extracts the majority of the features directly from the corpus, including the sentence alignment, POS tags, lemmatization and parsing tree for every sentence. The pre-processing operations needed are mostly used to filter the sentences, in order to obtain a high quality in the candidate selection. Other than that, external components are used to align the sentences at the word level and to estimate the target proficiency level for the exercise.

Finally, the tokens are clustered to propose bilingual sentence reconstruction as a new type of exercise. In this game-like exercise the user is asked to link the words of a sentence in the target language to group of words in a chosen language, ideally the/a native language. It is based on the intuition, already experimented in research, that groups of words help the learner recognize language patterns, favouring the language acquisition process, other than improve the accuracy of the word alignment, for example in the case of function words.

The system is almost entirely language independent, and every component can be substituted without modifications to other parts of the system. By adding a complexity estimator component for another language, the implementation can theoretically be used with any language pair.

# 6 FUTURE WORK

There are plans to integrate the presented research in the platform Lärka. This would make it possible to collect real usage data, both to test the validity of the proposed exercise for language learning and to analyse the response of the learners to the subtitle sentences.

This is fundamental because while the annotations can tell us if the selected sentences are "good" translations or if the alignments are correct, it is the users' interactions which provide valuable feedback to decide whether or not a system can be applied in schools (see ICALL blog post[27]).

The system can also be extended to include other language pairs. As was said, the major part of the code is language independent, but it is necessary to include a new component to estimate the complexity of the sentences for the new languages, and to run more tests to see if the pre-processing operations are enough to obtain an acceptable accuracy in the results.

In future implementations, the sentence selection process could be improved to consider the semantics of the tokens, which has been ignored in this work, for example to check if the meaning of two aligned tokens is in the same semantic space. This would allow to automatically exclude all those sentences in which the structure is similar (i.e. NOUN + VERB + NOUN in both languages) but the meaning is different.

The exercise could also be extended to be more elaborate. At lower proficiency levels, it could give suggestions to the user, colouring the tokens with the same POS tag; at higher proficiency levels, instead of the word in the correct form, it could show its lemma and ask the learner to change it, either by selecting a form from a list of possibilities or with a free text entry.

Finally, the system could collect usage information to create a profile for the learners and recommend increasingly difficult exercises, following their progresses.

[27] https://spraakbanken.gu.se/blogg/index.php/2020/04/30/common-pitfalls-in-the-development-of-icall-applications/

# REFERENCES


*Aprosio A., Moretti G. (2018), "Tint 2.0: an All-inclusive Suite for NLP in Italian", Accademia University Press, ID: 10.4000/books.aaccademia.3571*

*Barzilay R., Lapata M. (2008), "Modeling Local Coherence: An Entity-Based Approach", Computational Linguistics, Volume 34, Number 1, March 2008, pages 1-34, https://www.aclweb.org/anthology/J08-1001*

*Borin L., Forsberg M., Lonngren L. (2013), "SALDO: a touch of yin to WordNet's yang.", Language Resources and Evaluation, 47(4):1191–1211.*

*Boulton A. (2017), "Integrating corpus tools and techniques in ESP courses", ASP [Online], 69 | 2016, URL : http://asp.revues.org/4826*

*Bourdaillet J., Huet S., Langlais P., Lapalme G. (2010), "TransSearch: From A Bilingual Concordancer To A Translation Finder", Mach. Transl., 24(3-4):241–271*

*Briscoe T., Carroll J. (1995), "Developing and Evaluating a Probabilistic LR Parser of Part-of-Speech and Punctuation Labels", Proceedings of the Fourth International Workshop on Parsing Technologies, pages 48–58, https://www.aclweb.org/anthology/1995.iwpt-1.8*

*Brown P., Della Pietra S., Della Pietra V., Mercer R. (1993), "The Mathematics of Statistical Machine Translation: Parameter Estimation", Computational Linguistics, Volume 19, Number 2, June 1993, Special Issue on Using Large Corpora: II, pages 263–311, https://www.aclweb.org/anthology/J93-2003*

*Byrd P., Coxhead A. (2010), "On the other hand: Lexical bundles in academic writing and in the teaching of EAP", University of Sydney Papers in TESOL, 5, pages 31-64*

*Chiang D. (2005), "A hierarchical phrase-based model for statistical machine translation", Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL), pages 263-270*

*Cobb T., Boulton A. (2015), "Classroom applications of corpus analysis". In D. Biber & R. Reppen (eds), Cambridge Handbook of English Corpus Linguistics. Cambridge: Cambridge University Press, pages 478-497. DOI: 10.1017/CBO9781139764377.027*

*Cobb T. (1999), "Applying Constructivism: A Test for the Learner-as-Scientist", Educational Technology Research and Development, 47(3), 15-31. DOI: https://doi.org/10.1007/BF02299631*

*Council of Europe (2001), "Common European Framework of Reference for Languages: learning, teaching, assessment", Cambridge, Cambridge University Press*

*Dell'Orletta F., Montemagni S., Venturi G. (2011), "READ-IT: assessing readability of Italian texts with a view to text simplification", Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies (SLPAT 2011), pages 73–83*

*DeNero J., Klein D. (2007), "Tailoring Word Alignments to Syntactic Machine Translation", Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 17-24, https://www.aclweb.org/anthology/P07-1003*

*Dodigovic, M. (2005), "Artificial Intelligence in Second Language Learning: Raising Error Awareness", Clevedon: Multilingual Matters LTD*

*Dürlich, L., François, T. (2018), "EFLLex: A Graded Lexical Resource for Learners of English as a Foreign Language", Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018), pages 1–7*

*Dyer C., Chahuneau V., Smith N. (2013), "A simple, fast, and effective reparameterization of IBM Model 2", Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 644–648*

*François T., Volodina E., Ildiko Pilán, and Tack A. (2016), "Svalex: a CEFR-graded lexical resource for Swedish foreign and second language learners", Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), pages 213–219*

*Gale W., Church K. (1993) "A Program for Aligning Sentences in Bilingual Corpora," Computational Linguistics, 19:1, pp. 75-102*

*Galley M., Graehl J., Knight K., Marcu D., DeNeefe S., Wang W., Thayer I. (2006), "Scalable inference and training of context-rich syntactic translation models", Proceedings of ACL 2006*

*Graën J., Alfter D., Schneider G. (2020), "Using Multilingual Resources to Evaluate CEFRLex for Learner Applications", Proceedings of The 12th Language Resources and Evaluation Conference, pages 346-355, https://www.aclweb.org/anthology/2020.lrec-1.43*

*Granger S. (2008), "Learner corpora in foreign language education", Van DeusenScholl, N. and Hornberger, N. H. (eds.), Encyclopedia of Language and Education, 24, Volume 4: Second and Foreign Language Education, New York: Springer, pages 337–51*

*Gray J. (2010), "The Construction of English: Culture, Consumerism and Promotion in the ELT Global Coursebook", Palgrave Macmillan*

*Halacsy P., Kornai A., Oravecz C. (2007), "Hunpos – an open source trigram tagger", Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, pages 209–212*

*Hamburger H. (1995), "Tutorial Tools for Language Learning by Two-Medium Dialogue", V. M. Holland, J. D. Kaplan, M. R. Sams, editors, Intelligent*

*Language Tutors: Theory Shaping Technology. Lawrence Erlbaum • Associates, Mahwah, New Jersey*

*Heift T., Schulze M. (2007), "Errors and Intelligence in Computer-Assisted Language Learning: Parsers and Pedagogues", New York: Routledge (Routledge series in computer-assisted language learning, edited by Carol Chappelle), 2007, xviii+283 pp; hardbound, ISBN 978-0-415-36191-0*

*Higgins J. (1987), "CALL evaluation tools", Report of the St Martin's Conference (September 1986)*

*Johns T., King P. (1991), "Classroom Concordancing", English Language Research Journal, 4, Centre for English Language Studies, the University of Birmingham, 1991, pages 178*

*Kilgarriff A., Husák M., McAdam K., Rundell M., Rychlý P. (2008), "GDEX: Automatically finding good dictionary examples in a corpus", Proceedings of the 13th EURALEX International Congress. Spain, July 2008, pages 425–432*

*Kincaid J., Fishburne R., Rogers R., Chissom B. (1975), "Derivation of new readability formulas for Navy enlisted personnel", Technical report, Branch Report 8-75, United States, Naval Education and Training Support Command, Chief of Naval Technical Training*

*Klimova B., Kacet J. (2017), "Efficacy of Computer Games on Language Learning", TOJET: The Turkish Online Journal of Educational Technology – October 2017, volume 16 issue 4*

*Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran Richard Zens C., Dyer C., Bojar O., Constantin A., Herbst E. (2007), "Moses: Open source toolkit for statistical machine translation", Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic*

*Laufer B., Ravenhorst-Kalovski G. (2010), "Lexical threshold revisited: Lexical text coverage, learners' vocabulary size and reading comprehension", Reading in a Foreign Language, Volume 22, No. 1, pages 15–30, ISSN 1539-0578*

*Lawson A. (2001), "Collecting, aligning and analysing parallel corpora", in "Small Corpus Studies and ELT: Theory and practice", pages 279-. DOI: https://doi.org/10.1075/scl.5.17law*

*Leech G. (1997), "A Brief Users' Guide to the Grammatical Tagging of the British National Corpus", UCREL, Lancaster University*

*Levy, M. (1997), "Computer-assisted language learning: Context and conceptualization", Oxford: Clarendon, DOI:10.2307/417519*

*Lison P., Tiedemann J. (2016), "OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles", Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 923–929, https://www.aclweb.org/anthology/L16-1147*

*Lison P., Tiedemann J., Kouylekov M. (2018), "OpenSubtitles2018: Statistical Rescoring of Sentence Alignments in Large, Noisy Parallel Corpora",*

*Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), https://www.aclweb.org/anthology/L18-1275*

*Lucisano P., Piemontese M. (1988), "GULPEASE: una formula per la predizione della difficolta dei testi in lingua italiana", Scuola e città, 3(31), pages 110–124*

*Manning C., Surdeanu M., Bauer J., Finkel J., Bethard S., McClosky D., "The Stanford CoreNLP Natural Language Processing Toolkit", Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55–60, DOI: 10.3115/v1/P14-5010*

*Fox J., Matthews C. (1991), "Learner strategies and learner needs in the design of CALL help systems", Proceedings of EUROCALL, Helsinki, pages 127-132*

*Matthews C. (1993), "Grammar frameworks in intelligent CALL", CALICO Journal, 11(1), pages 5–27*

*Minack E., Siberski W., Nejdl W. (2011), "Incremental diversification for very large sets: a streaming-based approach", Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, SIGIR '11, pages 585–594, New York, NY, USA*

*Mishan F., Strunz B. (2003), "An application of XML to the creation of an interactive resource for authentic language learning tasks", Cambridge University Press, DOI: https://doi.org/10.1017/S095834400300082X*

*Nerbonne J., Heeringa W. (2002), "Measuring dialect distance phonetically", Proceedings of SIGPHON-97: Meet. ACL Special Interest Group Computational Phonology, Madrid (3), Spain, http://www.cogsci.ed.ac.uk/sigphon*

*Nivre J., Nilsson J., Chanev A., Eryiğit G., Kübler S., Marinov S., Marsi E. (2007), "MaltParser: A language-independent system for data-driven dependency parsing", Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), pages 2216–2219*

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

*Östling R., Tiedemann J. (2016), "Efficient Word Alignment with Markov Chain Monte Carlo", The Prague Bulletin of Mathematical Linguistics (PBML), Number 106, pages 125–146*

*Oxford R. (1993), "Intelligent Computers for Learning Languages: The View for Language Acquisition and Instructional Methodology", Computer Assisted Language Learning, 6(2), pages 173–188*

*Paetzold G., Specia L. (2016), "SemEval 2016 Task 11: Complex Word Identification", Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), pages 560–569, DOI: 10.18653/v1/S16-1085*

*Pasero R., Sabatier P. (1998), "Linguistic Games for Language Learning: A Special Use of the ILLICO Library", Computer Assisted Language Learning, 11(5), pages 561–85*

*Pianta E., Girardi C., Zanoli R. (2008), "The TextPro Tool Suite", Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, Marrakech, Morocco*

*Pilán I. (2013), "NLP-based approaches to sentence readability for second language learning purposes", Master's Thesis, University of Gothenburg. https://www. academia. edu/6845845/NLP-based_Approaches_to_ Sentence_Readability_for_Second_ Language_Learning_Purposes*

*Pilán I., Volodina E., Borin L. (2017), "Candidate sentence selection for language learning exercises: from a comprehensive framework to an empirical evaluation", arXiv preprint arXiv:1706.03530*

*Pilán I., Volodina E., Johansson R. (2013), "Automatic selection of suitable sentences for language learning exercises", 20 Years of EUROCALL: Learning from the Past, Looking to the Future: 2013 EUROCALL Conference Proceedings, pages 218-225*

*Pilán I., Volodina E., Zesch T. (2016), "Predicting proficiency levels in learner writings by transferring a linguistic complexity model from expert-written coursebooks", Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2101-2111*

*Potter A. (2004), "Interactive rhetoric for online learning environments", The Internet and Higher Education, vol. 7, pages 183-198, 2004/0/3rd 2004*

*Reeder F., Hamburger H., Schoelles M. (1999), "More Intelligent CALL", K.Cameron (Ed.), Computer-Assisted Language Learning, pages 183–202, Lisse: Swets & Zeitlinger*

*Salaberry R. (1996), "A Theoretical Foundation for the Development of Pedagogical Tasks in Computer-Mediated Communication", CALICO Journal, 14(1), pages 5–34*

*Schwarm S., Ostendorf M. (2005), "Reading level assessment using support vector machines and statistical language models", Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 523–530, Association for Computational Linguistics*

*Segler T. (2007), "Investigating the selection of example sentences for unknown target words in ICALL reading texts for L2 German", PhD Thesis, University of Edinburgh*

*Suphawat P., Dodigovic M. (1999), "Learning English on the Web within a global EAP community", Elm Bank Publications, Exeter, CALL and the Learning Community, pages 127-136*

*Teubert W., "Comparable or Parallel Corpora?" (1996), International Journal of Lexicography, Volume 9, Issue 3, September 1996, Pages 238–264*

*Tiedemann J. (2003), "Recycling translations: Extraction of lexical data from parallel corpora and their application in natural language processing", Acta Universitatis Upsaliensis, Studia Linguistica Upsaliensia 1. 130 pp. Uppsala. ISBN 91-554-5815-7*

*Tiedemann J. (2007), "Improved Sentence Alignment for Building a Parallel Subtitle Corpus: Building a Multilingual Parallel Subtitle Corpus", LOT Occasional Series 7, pages 147-162, Netherlands Graduate School of Linguistics*

*Tiedemann J. (2008), "Synchronizing Translated Movie Subtitles", Proceedings of the 6th International Conference on Language Resources and Evaluation, LREC 2008, Marrakech, Morocco*

*Tiedemann J. (2012), "Parallel Data, Tools and Interfaces in OPUS", Proceedings of the Eight International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, European Language Resources Association (ELRA), ISBN 978-2-9517408-7-7*

*Tonelli S., Manh K., Pianta E. (2012), "Making readability indices readable", Proceedings of the First Workshop on Predicting and Improving Text Readability for target reader populations, pages 40–48, Montr´eal, Canada, June. Association for Computational Linguistics*

*van der Linden, E. (1993), "Does Feedback Enhance Computer-Assisted Language Learning?", Computers and Education, 21(1–2), pages 61–65*

*Volodina E., Kokkinakis S., Johansson R. (2012), "Semi-automatic selection of best corpus examples for Swedish: Initial algorithm evaluation", Workshop on NLP in Computer-Assisted Language Learning. Proceedings of the SLTC 2012 workshop on NLP for CALL. Linköping Electronic Conference Proceedings, pages 59–70*

*Volodina E., Pijetlovic D., Pilán I., Kokkinakis S. (2013), "Towards a gold standard for Swedish CEFR-based ICALL", Proceedings of the Second Workshop on NLP for Computer-Assisted Language Learning. NEALT Proceedings Series, 17*

*Volodina E., Pilán I., Borin L., Tiedemann T. (2014), "A flexible language learning platform based on language resources and web services", Proceedings of LREC 2014, Reykjavik, Iceland, pages 3973-3978*

*Wu S., Fitzgerald A., Witten I. (2014), "Second Language Learning in the Context of MOOCs", Proceedings of the 6th International Conference on Computer Supported Education, volume 1 April 2014, pages 354–359*

*Zariņa I., Ņikiforovs P., Skadiņš R. (2015), "Word alignment based parallel corpora evaluation and cleaning using machine learning techniques", Proceedings of the 18th Annual Conference of the European Association for Machine Translation, Antalya, Turkey, pages 185–192*

*Zenkel T., Wuebker J., DeNero J. (2020), "End-to-End Neural Word Alignment Outperforms GIZA++", Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1605–1617, Association for Computational Linguistics, DOI: 10.18653/v1/2020.acl-main.146*

# APPENDIX

The complete code of the project is available through GitHub: https://github.com/Ari-Zanetti/Thesis