Imperial College London

INDIVIDUAL PROJECT INTERIM REPORT

DEPARTMENT OF COMPUTING

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

Detecting Automatically Translated Documents

Author:
Calin-Andrei Alexandru

Supervisor:
Dr. Thomas Lancaster

April 9, 2021

Contents

1 Introduction **1**
- 1.1 The Problem . . . 1
- 1.2 The Proposed Solution . . . 2

2 Background **4**
- 2.1 CL Translation Detection Models . . . 4
- 2.2 CL Translation Detection Methods . . . 5
  - 2.2.1 CL-CNG . . . 6
  - 2.2.2 CL-CTS . . . 7
  - 2.2.3 CL-ESA . . . 10
  - 2.2.4 CL-ASA . . . 12
  - 2.2.5 Translation + Monolingual Analysis . . . 14
- 2.3 CL Translation Detection Methods Comparison . . . 16

3 Project Plan **21**
- 3.1 The End Product . . . 21
- 3.2 The Documentation . . . 21
- 3.3 The Milestones . . . 22
- 3.4 The Possible Extensions . . . 22
- 3.5 The Current Status . . . 22

4 Evaluation Plan **23**
- 4.1 The Website . . . 23
  - 4.1.1 Unit Testing . . . 23
  - 4.1.2 User Testing . . . 23
- 4.2 The Detection Algorithm . . . 24
  - 4.2.1 Recall . . . 24
  - 4.2.2 Precision . . . 24
  - 4.2.3 Conclusion . . . 24

5 Ethical Considerations **25**

Chapter 1

Introduction

Automatic translation, also known as machine translation is a ”translation carried out by a computer”. It is sometimes referred to as being a **Natural Language Processing** technique which builds language and phrase models used to translate text from other language assets [7]. Since its creation, machine translation rapidly advanced and in this current age there are plenty of useful and powerful services such as **Google Translate** that provide their users with automatic translation mechanisms. When automatic translation tools first appeared, it was easier to distinguish a manually produced translation from one produced by a machine since the latter was less accurate than the former. However, this has changed and along with it, issues such as cross-language plagiarism in academic institutions appeared, that were not encountered beforehand have emerged. This project will aim to tackle the aforementioned issue with the help of an algorithm that detects automatically translated texts.

1.1 The Problem

Plagiarism has been an important issue for academic institutions since their inception, but it became even more common ever since the rise of the World Wide Web. This happened because the internet grew rapidly and it currently contains an impressive collection of resources [9]. Plagiarism can take many forms, from extracting and paraphrasing information from another person's work without giving credit, to straight out copying parts or even another person's entire work. However, in recent years, a new form of plagiarism has emerged, a form which is hard or even impossible to automatically detect: cross-language plagiarism or translation plagiarism [16].

This form of cheating occurs when one takes work written in a different language and translates it into English manually or uses automatic translation services such as **Google Translate**. Since academic submissions are generally passed through a plagiarism check, individuals who engage in cross-language plagiarism try to cheat this filter, which might not signal any issues for work that has been translated from another language, since during this process some form of rewriting is involved.

In addition to facilitating cross-language plagiarism, automatic translation services

may also lead to people who hire professionals to produce a translation for a document not receiving the service that they paid for. Since the client might not know the language of translation, that would lead to them not having a way of knowing whether the translation was performed automatically or manually. The client could try and translate the document back from English into the original language, but it would be nearly impossible to conclude if the differences are not obvious enough.

For example, if the following sentence from Romanian is taken: ”Clientul nu are cum să știe dacă a fost păcălit sau nu”, translated into English and then back into Romanian, the result will be: ”Clientul nu are de unde să știe dacă a fost păcălit sau nu”. The semantic of the sentence is the same, the only notable difference being a small syntactic change, which comes from replacing one of the words in the original sentence with a synonym. Therefore, in this specific example, there is no obvious reason to suspect the use of automatic translation. But perhaps, at a closer inspection, certain machine translation patterns might be uncovered, patterns that would make the process of detection of automatically translated documents more effective.

1.2 The Proposed Solution

It becomes clear that both the problem of cross-language plagiarism in academic institutions and human translation services not delivering what was requested of them might arise from the use of automatic translation software. Therefore, the project aims to create a mechanism that allows people to easily check whether a piece of text has been automatically translated or not. Ideally, this mechanism will be hosted on a website that can be accessed by any person and it will be intuitive and easy to use.

The mechanism that will be at the core of this website will combine **machine learning** techniques, more specifically *Supervised Learning* with **Natural Language Processing** techniques, such as *Stemming* and *Machine Translation*. The mechanism will take the form of a *Neural Network* which will perform a classification task since it has to answer to the question of whether a text was automatically translated or not. The main idea would be to make use of several cross-language datasets available online, which contain numerous documents that have been manually translated by professionals. The algorithm will be provided with thousands of examples of both manually and automatically translated documents with the end-goal of being able to correctly classify texts that it has not previously seen into either machine translated or human translated.

Since the publicly available datasets contain a vast number of data points, the data will probably be split into three sets: one for training, one for validation and one for testing, with an 80-10-10 ratio. The largest part of the data will be used for training, to create a model. The validation dataset will be used for hyperparameter tuning, to improve the performance of the previously generated model, while the testing one will be used for determining the accuracy on previously unseen data. The use of a

training dataset which contains data points that have not been used for training or validation will provide a general idea on the performance of the model, ensuring its robustness and that it does not overfit to the training data. The goal will be to adjust the design in a way that maximizes the accuracy while minimizing the computation time.

Since this website will be part of the public domain and available for use by any person, members of the academic community will have at their disposal a mechanism that will allow them to perform an extra check for plagiarism and people that hire a professional for a translation service will have a way of checking whether or not what they paid for was justified.

Chapter 2

Background

In this section, an overview of the different *State-of-the-Art* methods will be presented. These mechanisms are not used for directly detecting translation plagiarism, but to detect textual similarities between documents or fragments of them, from two different languages. There are five retrieval models that present different approaches for performing the task of textual similarity detection. Each of them has the same goal, of deciding whether or not two or more pieces of text written in different languages have the same meaning [1].

2.1 CL Translation Detection Models

In this section, the five aforementioned models will be presented, to offer a general idea regarding the different approaches that can be taken for combating cross-language plagiarism.

- **Syntax-Based** models distinguish themselves by the fact that multilingual documents can be compared without the need of being translated, by just looking at the syntactic structure of the sentences. The best performance is obtained when working on pairs of languages that share a similar syntactic structure, for example, the *Romance* languages [5]. The most common approach of this class is the *Cross-Language Character N-Gram* method or in short **CL-CNG**.

- **Dictionary-Based** models make use of thesauri, dictionaries or other concept spaces [5], such as the *JRC-Acquis Multilingual Parallel Corpus* developed by *Eurovoc*, which establishes connections between texts through *language-independent anchors*. Those *anchors* are pairs of words from numerous languages denoting entity names, locations, dates etc [2]. The most common approach of this class is the *Cross-Language Conceptual Thesaurus-Based Similarity* method or in short **CL-CTS**.

- **Comparable Corpora-Based** models look into comparing the bodies of texts. Unlike **Parallel Corpora-Based** models, **Comparable Corpora-Based** models do not make use of sentence-aligned translations. However, this concept is best illustrated through examples such as **Wikipedia** and similar data sources,

by taking advantage of the texts with the same topic that have a common vocabulary. Although noisier than the parallel model, the comparable corpora is more flexible [5]. The most common approach of this class is the *Cross-Language Explicit Similarity Analysis* method or in short **CL-ESA**.

- **Parallel Corpora-Based** models make use of sentences being aligned, as mentioned above, by use of the **IBM Model 1** [10]. In addition to this, the position of the words is taken into account and statistical dictionary probabilities are calculated with the help of an *expectation-maximization* algorithm [5]. The most common approach of this class is the *Cross-Language Alignment Similarity Analysis* method or in short **CL-ASA**.
- **MT-Based** models first determine the language from which the suspicious text was translated from with the help of a language detector. Afterwards, the fragment is translated and monolingual analysis is performed [5]. The most common approach of this class is the *Translation + Monolingual Analysis* method or in short **T+MA**.

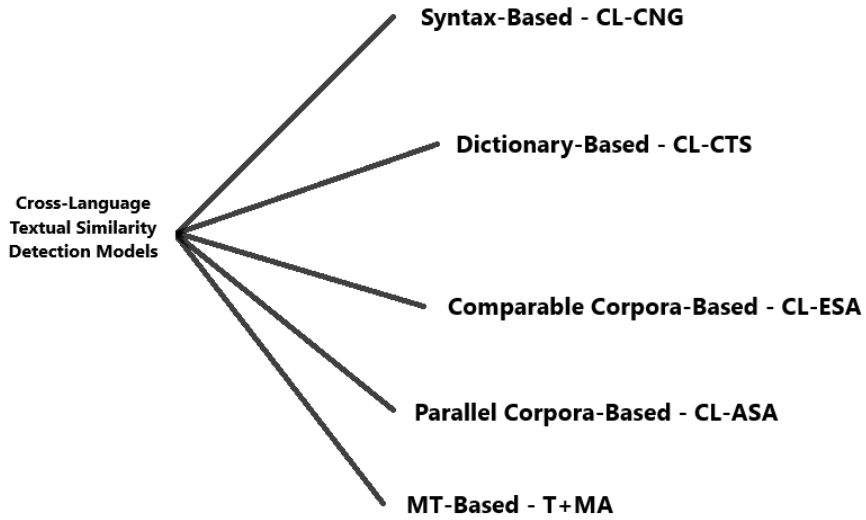

Figure 2.1: The different retrieval models

2.2 CL Translation Detection Methods

As mentioned above, there are several techniques that can be used for performing the task of cross-language translation detection. However, this section will present one approach from each aforementioned model, alongside the approaches that could be most relevant to the development of the algorithm mentioned in the Introduction.

The most important step of cross-language translation detection is *Information Retrieval*, which given two documents, d_1 and d_2, compares them with the help of a *Retrieval Model* R, which is a tool for computing representations D_1 and D_2 of the aforementioned documents, alongside a *similarity function* f. This function receives as parameters D_1 and D_2 and its result is a real number, which offers information about the textual similarity between d_1 and d_2 [12]. One of the most commonly used functions is the cosine similarity, which returns a value in the interval $[0, 1]$. The closer the value is to one, the higher the similarity between the two given documents is.

2.2.1 CL-CNG

As its name suggests, the *Cross-Language Character N-Gram* approach makes use of *n-grams*, which are a reunion of n arbitrary chosen characters. Due to the fact that a dictionary is needed for information retrieval and the size of it may be very large or in theory even infinite, a method that would lead to its effective reduction had to appear. Once the n-gram approach was applied to the task of information retrieval, the number of items in the collection drastically reduced, since it was bounded by the number of letters in the alphabet to the power of n. Therefore, for an n that is quite small, in the range $[1, 5]$ for example, the number of possible *n-grams* becomes tractable in the context of memory handling [6].

For example, in the case of the Romanian alphabet, which contains 32 characters, including the whitespace and *2-grams*, the size of the dictionary would be 1024, which is quite a small number. In this same case, no more than 32.768 *3-grams* could be found. Although a significantly larger number, it is still not a big challenge, computationally-wise for most modern-day CPUs.

CL-CNG is the oldest, being developed in 2004, and arguably the simplest of the models described in this section. Since the variant that produces the best results is **CL-C3G**, from this point forward, *3-grams* will be used as the standard when offering details about this approach. It uses a simplified alphanumeric alphabet, which does not contain any special characters or symbols. As a result, before performing the separation into *3-grams*, any given text must first have all its characters transformed into lowercase and have its punctuation marks, blank spaces and symbols deleted. After all these preliminary steps have been performed, the text can be separated into *3-grams* [12].

In other words, the given document is represented as a vector with a dimension of the size of the alphabet to the power of three. Since in a given text only a part of all the possible *3-grams* occur, the vector space can be sparsely populated. Because of this, its elements have to be weighted in accordance with a standard weighting scheme, which in this case is the *term frequency-inverse document frequency* or in short **TF-IDF** [12]. This metric is calculated by multiplying two different mea-

surements: term frequency and inverse document frequency. The former relates to simply counting the number of occurrences of a given word in a text and possibly adjusting this number to the length of the document. The latter relates to calculating how rarely or often is a word encountered in a given set of documents. **TF-IDF** takes real values in the interval $[0, 1]$. If the number is very close to 0, it implies that the word is common and comes up in numerous documents [19].

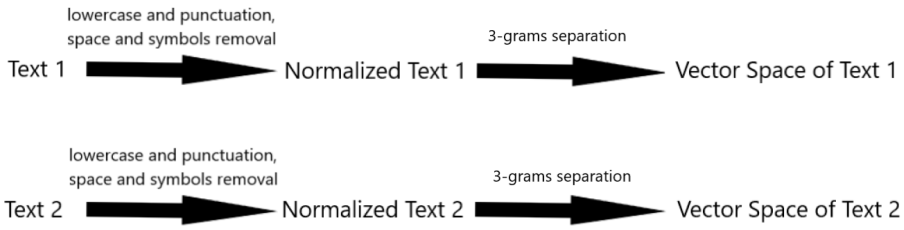

Figure 2.2: The first steps of the CL-C3G approach

After these steps are performed for the two candidate texts, the only part remaining is to assess the textual similarity. This is done using the *cosine similarity* measure since the documents have been previously represented as vector spaces. The advantage of using this metric is that it doesn't take into account the size of the documents when computing the similarity of two documents. From a mathematical point of view, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. In the case of **CL-C3G**, the two vectors are the arrays that contain the *3-grams* of the two documents. If plotted on a multi-dimensional space, with each dimension being represented by a *3-gram*, the *cosine similarity* will showcase the angle of the words, rather than their dimension. This is preferable in the given case due to the fact that a smaller angle will result in a higher similarity between the two documents [14].

In conclusion, **CL-C3G** is a simple, but powerful approach for computing textual similarity between two given texts. However, in some certain cases, it might be too slow. In addition to this, smaller documents have fewer words, which results in a low number of *3-grams*. Therefore, most *3-grams* in the dictionary will have a value of 0 when constructing the vector space which will lead to inefficient use of memory.

2.2.2 CL-CTS

The *Cross-Language Conceptual Thesaurus-Based Similarity* method aims to calculate textual similarity of two given documents by a measure of *shared concepts*, which is assigned using a **Conceptual Thesaurus** and **Named Entities** [11].

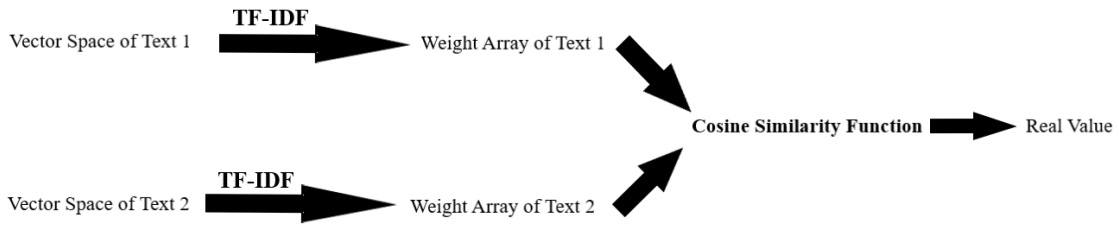

Figure 2.3: The final steps of the CL-C3G approach

A **Conceptual Thesaurus** incorporates words or structures of words and tries to cover all the concepts of various domains. One of the most popular and useful CTs is *Eurovoc*, which was mentioned in the previous section. It was created by the European Parliament and is currently maintained and constantly updated by the Publications Office of the European Union, with new words or languages [11]. It contains more than 6000 multilingual concepts which have been categories through identification numbers. In addition to this, the thing that makes it valuable is that the words that it contains have been translated into 27 languages and include more than 20 domains that the European Parliament activates in.

English	Romanian	French	German
commercial law	drept comercial	droit commercial	Handelsrecht
foreign policy	politică externă	politique extérieure	Außenpolitik
working conditions	condiții de muncă	condition de travail	Arbeitsbedingungen

Table 2.1: Multilanguage samples extracted from Eurovoc

Similar to **CL-C3G**, the documents that are checked for textual similarity are represented as vectors. However, instead of creating a vector space from the words in the documents, this approach creates it using the concepts present in *Eurovoc*. However, the problem of word assignment is not easily solvable, since directly assigning a word based on its direct occurrence proved to produce insignificant results [3]. Due to this, a concept is assigned to a document through the following function:

$$v(e,d) = \sum_{t \in e, T_e} f(t,d) \tag{2.1}$$

In this function, e is an *Eurovoc* concept and d is one of the documents that need to be analyzed. The function f is used for calculating how frequently t, the stemming

of e, appears in the text and T is the set of all *Eurovoc* words. A given concept e is assigned to document d, with the weight v(e,d), if the function v(e,d) produces a value larger than 0 [11].

Due to its multilingual structure, the CT can be exploited based on the idea that 2 or more terms combined can be domain-dependent, but each term alone can be domain-independent. For example, a construction such as ”working conditions” is more specific than just the terms ”working” or ”conditions”, which may be present in a vast number of domains. As a result, not all terms are useful when computing the similarity estimation between two given documents. The function of document frequency or *df* was introduced to combat the noise introduced by the similarity increase of non-relevant documents when df(t) is very high. A new set was created from T, in short **RC**, from reduced concepts, which includes only those terms which satisfy the following equation, with the variable β being an arbitrarily small value [11]:

$$0 < df(t) < \beta \tag{2.2}$$

In a previous paragraph, it was mentioned that the concepts in the *Eurovoc* CT receives an identification number. Due to this, when constructing the vector space of the document, each id represents one dimension. to compute the textual similarity between two given documents d_1, in language L_1 and d_2, in language L_2, the following equation is used [11]:

$$\omega(d1, d2) = \frac{\alpha}{2} * \left(\frac{\overrightarrow{c_{d1}} \cdot \overrightarrow{c_{d2}}}{|d1||d2|} + l(d1, d2) \right) + (1 - \alpha) * \xi(d1, d2) \tag{2.3}$$

This equation contains two terms: the CT component and the *named entity* component. The c vectors correspond to the conceptual vectors of the two documents that are compared, while $||$ denotes the size in words of d_1 and d_2. The *xi* function returns the *cosine similarity* of the character *3-grams* of the *named entities* of the two documents, while the function *l* is called the length factor penalty for the given documents and is defined in [13]. $(1 - \alpha)$ is a weighting factor used to normalize the output of the function between the interval [0,1] [11].

The use of *named entities* and implicitly of the second component of the equation emerged from the idea that those **NEs** can be considered as distinguishing features when identifying diverse documents on the similar conceptual topics. Since the *named entities* usually have more or less significant variations through different languages, *character n-gram* is used to compute textual similarity. In addition to this, since parallel documents have specific length distributions [13], which helps with the process of integrating the information about length for parallel document pairs. Therefore, by using the *length factor*, this information is induced in the textual similarity estimation process [11].

In conclusion, although **CL-CTS**, outperforms **CL-CNG** from a computational and temporal point of view, it produces better results only on particular datasets. However, its strongest advantage is that it produces high stability across all the datasets while performing consistently [11].

2.2.3 CL-ESA

The *Cross-Language Explicit Similarity Analysis* method is a multilingual retrieval model used for computing cross-language textual similarity. This approach makes use of the multilingual alignment of Wikipedia documents. The main goal of this method is to generate two vectors, for two documents written in two different languages, and to compare them using a measure such as the cosine similarity. The most important feature of this retrieval model is that it performs semantic analysis without the need for automatic translation capabilities. The precursor of this approach is the retrieval model called *Explicit Semantic Analysis*, or in short **ESA** [8].

The idea of **ESA** is to use a collection of documents to encode the specific knowledge of a given document with respect to it. Therefore, for each specific document in the set, a single concept is created, which the given text is compared to. Due to this, the *Cross-Language Explicit Similarity Analysis* approach represents documents as n-dimensional concept vectors using the following formula [8]:

$$\mathbf{d} = \left(\varphi(v, v_1^*), \varphi(v, v_2^*), ..., \varphi(v, v_n^*)\right)^T \tag{2.4}$$

This approach uses a collection of texts D^* called index documents, which has a size of n. From D^*, v_i^* is extracted, representing the vector space model representation corresponding to the *i*th entry in D^*. In addition to this, v is the vector space model representation for the document that needs to be analyzed. The dimensional vector **d** is created by applying the *cosine similarity* function φ to all the (v,v_i^*) pairs. In the case in which the *cosine similarity* of such a pair is too small, the value is set to 0 [8].

After calculating the n-dimensional concept vectors **d1** and **d2** for two documents that need to be analyzed for textual similarity, the *cosine similarity* function is applied again, to **d1** and **d2** this time, with the result being defined as the similarity between the two documents under ESA [8].

The method presented above can naturally extend to multiple languages due to its nature, by having document index collections in different languages. Due to this, there is no need for any translation technology, just a comparable set of texts about similar topics written in various languages. One such set of documents is Wikipedia, which contains a variety of concepts written in many different languages and one such approach that makes use of it is **CL-ESA**. [12].

The *Cross-Language Explicit Similarity Analysis* method features 3 different sets: **L**, D^* and **C**. **L** represents the collection of languages, D^* is the set of index document collections, with each D_i^* containing texts in the language L_i and **C** contains several *concept descriptors*. If D^* has the property that *i*th document of all the index document collections in D^* describes c_i in the language l_i then it is called a *concept-aligned comparable corpus* [12].

Given two documents written in two different languages, they are both represented as **ESA** vectors with the help of the index document collections corresponding to the languages of the documents. The similarity analysis is performed in the concept space, by computing the cosine similarity between the vectors associated with the two documents. In this step, **CL-ESA** utilizes the reasoning of a *comparable corpus alignment* that if all the concepts in C are described well and complete for all the languages in the set **L**, the two documents are represented in *comparable concepts spaces*, with the use of the index document collections associated to them. For this approach to yield results, each index document set must meet the requirements of **ESA** [12].

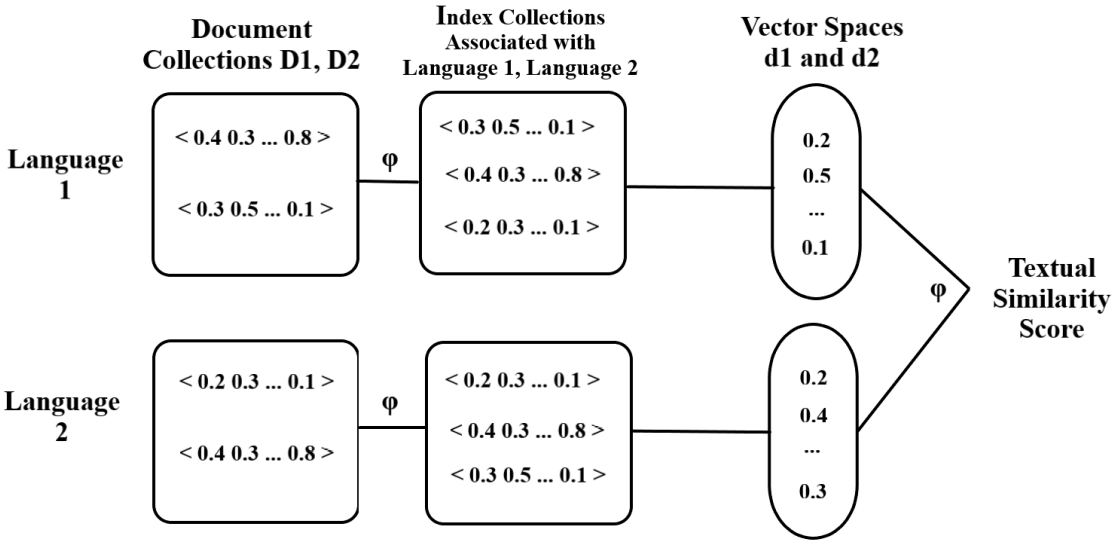

Figure 2.4: CL-ESA approach

The *Cross-Language Explicit Similarity Analysis* approach can be tuned to obtain a higher or lower retrieval quality, depending on the number of desired languages. For example, if the former is desired, the documents that need to be analyzed have to be represented as 10^5-dimensional concept vectors. However, in this case, the downside of having high accuracy is that computation time becomes quite significant and no more than 2 languages can be represented at the same time. If high multilingualism is needed, the texts should be represented as 10^3-dimensional concept vectors, for the approach to be feasible from a computational point-of-view. However, the retrieval quality is highly affected by this representation. A decent balance between

computational time and retrieval quality is achieved by representing the documents as vector concept spaces with dimensions between 10^3 and 10^4 [12].

2.2.4 CL-ASA

As mentioned before. **CL-ASA** has at its core statistical machine translation technology. It initially functioned by combining two measures, the *translation model probability* and *language model probability*. However, those two models were replaced with ones that produced better results: the *Translation Model* and *Length Model*, which will be presented next [12].

The *Cross-Language Alignment Similarity Analysis* method entails the creation of a *bilingual statistical dictionary* called *Cross-Lingual Plagiarism Analysis system* or in short **CLiPA** on the assumption that the parallel corpus is aligned and using the *IBM Model 1* [5].

The *Model 1* is a probabilistic generative model that functions inside a benchmark which assumes that any given source sentence **S** of length **l** translates into a target sentence **T**. This translation is ensured through a process which consists of 2 steps. The first step is generating a length *m* for the target sentence. The second step is done for each target sentence position $j \in 1, ..., m$ and consists in selecting a generating word from **S** and creating a target word t_j at position *j* with respect to the generating word from **S** [10].

However, *Model 1* functions on a simpler variation of this framework, which assumes that all the possible lengths for the target sentence have a uniform probability ϵ, generating words for source sentence are equally likely and the translation probability, defined as $tr(t_j|s_i)$, of the target language generated word depends only depends on the generating word from the source language. The probability that a target sentence **T** is derived from the source sentence **S** is given by the following equation [10].

$$p(T|S) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} tr(t_j|s_i) \tag{2.5}$$

For the creation of a bilingual dictionary, *IBM Model 1* expects a *sentence-aligned parallel corpus*. Therefore, the probability of two texts being translated from one to another was defined as [12]:

$$p(d_1|d_2) = \prod_{x \in d_1} \sum_{y \in d_2} p(x, y) \tag{2.6}$$

In this formula, $p(x, y)$ represents the probability that the word x is a translation of the word y. Although it generated good results for translating sentences, **CL-ASA** has to function for entire texts, which present various lengths. Due to this, the model

was adapted to use weights instead of probabilities, changing from the *Translation Model Probability*, to simply the *Translation Model* [12]:

$$\omega(d_1|d_2) = \sum_{x \in d_1} \sum_{y \in d_2} p(x, y) \tag{2.7}$$

The other important component of the *Cross-Language Explicit Similarity Analysis* approach is the *Length Model*, which is based on the expectation that a pair of translated documents d_1 and d_2 are unlikely to have the same length, but their sizes should be closely related by a length factor for every language pair. Due to this, the model has been defined for a pair of documents d1 and d2 as [12]:

$$\varrho(d_2) = exp\left(-0.5\left(\frac{|d_2|/|d_1| - \mu}{\sigma} \right)^2 \right) \tag{2.8}$$

The variables μ and σ represent the average and the standard deviation of the character lengths produced by document translations from the language of d_1 to the language of d_2, while $|d_1|$ and $|d_2|$ represent the lengths of the two texts [12].

As mentioned in the beginning of the subsection, **CL-ASA** was initially based on a *translation model probability*, written as $p(d_1|d_2)$ and *language model probability*, written as $p(d_2)$. Given a document d_1 written in the language l_1 and a document d_2 from a collection of documents $D^{'}$ written in language $L^{'}$ the approach measured the probability that d_2 is a translation of d_1, using Bayes' rule [12]:

$$p(d_2|d_1) = \frac{p(d_2)p(d_1|d_2)}{p(d_1)} \tag{2.9}$$

Since the purpose of the model is to retrieve the possible translation of d_1 for the target language $L^{'}$ and not to simply translate d_1 into $L^{'}$, the formula was adapted to use the *Translation Model* and *Length Model* instead. Therefore, the similarity of two given documents in the context of **CL-ASA** is now measured by the following model:

$$\varphi(d_1|d_2) = \varrho(d_2)\omega(d_1|d_2) \tag{2.10}$$

It is observed that this formula is not normalized, unlike other similarity measures, however, the partial order generated amid the documents matches the order of other similarity measures. From formula 2.7 it results that if valid translations from word x to word y appear in the language vocabularies, the weight increases. In addition to this, from 2.8 it can be observed that if the translation of document d2 from d1 does not have the expected length, the similarity measure reduces.

2.2.5 Translation + Monolingual Analysis

Unlike any of the previously described approaches, which only use principles of *Machine Translation* to compute the textual similarity between two documents, **T+MA** methods produce actual machine translations of one of the two given texts. This is the first step of the process and it is performed to reduce the complexity of the problem, by transforming it from multilingual into monolingual. This idea gained popularity quickly and therefore, different methods were proposed in the following years [4].

One such method is the *Language Normalisation*, which implies the creation of a representation where all the documents are written in the same language. This is considered to be one of the most popular preprocessing strategies for the information retrieval step of textual similarity detection. The language proposed as the base for this representation is English since the majority of the content available on the World Wide Web is written is available in this language and automatic translation mechanisms for a language pair are most likely to exist when English is one of the two languages [4].

The first step of this method is to determine which is the most likely language the two given documents were written in, with the help of a language detector. In the case in which one of the texts is not written in English, it is translated into it. The second step is to perform the textual similarity detection, which can be done through various methods since the process is monolingual. One such method, which is considered to be a good option due to its simplicity is the *Character n-Gram Profiles* or in short **CnGP** [4].

In **CnGP**, the two given documents are compared through their *profiles*, which are bags of *tf-weighted character 3-grams*. Different fragments s of the document are chosen and to compute their profiles, *sliding windows* of length m and step n are used. The dissimilarity between every profile p_s and p_d is computed with the aid of one of the normalised documents using the following formula [4]:

$$nd_1(p_s, p_d) = \frac{\sum_{t \in p_s} \left(\frac{2(tf_{t,p_s} - tf_{t,p_d})}{tf_{t,p_s} + tf_{t,p_d}} \right)^2}{4|p_s|} \tag{2.11}$$

In this formula $tf_{t,p}$ represents the normalised term frequency of *t*, which is a *character n-gram* of s or d, while $|p_s|$ represents the size of the profile of the fragment $s \in d$. The values of the formula are bound between 0 and 1, with the maximum similarity being achieved at 0. A potential case of plagiarism arises when a higher than expected standard deviation appears between the profile of a fragment and the mean with respect to all the possible fragment profiles p_s and p_d [4].

Another method is to use *Web-based cross-language models*, which builds upon the same principles of *Language Normalisation*. The first such model was developed in 2009, to detect plagiarism from Malay into English and made use of the *Google Translation* and search APIs [4]. The method features several steps: translation of the given document into English, removal of the stop words, stemming of the words, identification of similar texts in the collection of documents, comparison of similar pattern and displaying a summary of the result.

Since this is a **T+MA**-based model, the translation of the document into English is the first step. This is done to improve the effectiveness of the textual similarity detection process since the document collection that is used is the World Wide Web, which contains a vast number of texts in the English language. The Google Translate API was used for this task since it produces accurate translations and was freely distributed at that moment in time. After this step is completed, the process of removing *stop words* begins. This is done to reduce the complexity of the next steps since *stop words* do not add any value or meaning to the sentences of the documents. In addition to this, they make up almost 40 to 50 per cent of the number of words in a document collection and their removal would save computation time and space, while not hindering the effectiveness of the retrieval steps [17].

For example, the most common *stop words* include the, our, a, an, by, all, what, ever, do, is etc, and a sentence such as ”What is plagiarism?” will reduce to simply ”plagiarism?” if the process of removing *stop words* occurs.

After the removal of the *stop words* from the documents, the next step is to perform stemming on the remaining terms. This process is performed on all the words and consists of obtaining their roots by removing the prefixes and suffixes. This is done to improve the effectiveness of the information retrieval step by using the root terms for pattern matching. One of the most common algorithms used for stemming is *affix stripping*, which is based on a set of rules for removing the suffixes and prefixes of the words [17].

For example, one such rule for prefix removal could be: ”If the term begins with *auto*, remove the *auto*” and for suffix removal: ”If the word ends in *-al*, remove the *-al*”. As a result, after the process of *affix stripping* is completed, words such as ”autobiography” would reduce to ”biography” and words such as ”regional” would reduce to ”region”.

After the translation and preprocessing of the document is complemented, the monolingual analysis of the generated text follows. Since the original document has been translated into English, the algorithm uses the World Wide Web as the collection of documents that it works with, in particular the *Google AJAX Search API*. This search engine is used for looking up keywords and certain sentences from the suspect text and it returns the most similar documents from the World Wide Web. Moving forward, the actual analysis is performed between the given text and the similar doc-

uments returned by the search engine. This is done with the help of a *fingerprint matching technique*, which fragments the text into *character n-grams*, compares the fingerprints associated with the suspect documents and the documents in the corpus and then provides a textual similarity score [17].

2.3 CL Translation Detection Methods Comparison

In this section a comparison will be made between the models that are considered to be the most relevant for the development of the algorithm proposed by this individual project. **CL-CTS** was discarded because no API was found for *Eurovoc* or other *Conceptual Thesauruses*. In addition to this, **T+MA** approaches are also discarded due to the fact that they are computationally expensive, requiring a translation step and a preprocessing step before the actual textual analysis is performed.

The three approaches that will be compared are **CL-C3G**, **CL-ESA** and **CL-ASA**. This choice was made due to the fact that APIs that are publicly available exist for comparable and parallel document collections that are required by **CL-ESA**, such as *Wikipedia* and **CL-ASA**, such as the *JRC-Acquis Multilingual Parallel Corpus*, which contains legal documents from the European Union which have been translated and aligned in 22 different languages [20]. **CL-C3G** is selected for the comparison due to its nature of not being computationally expensive while also performing well on various document collections. *Wikipedia* is used by **CL-ESA** due to the fact that it contains documents in more than 200 languages which are also linked between one another when describing the same topic [8]. In 2010, an evaluation of these three models was performed. For both document collections, texts that do not have aligned versions in all of the languages required for the comparison are discarded and as a result 45.984 and 23.564 documents remain in the *Wikipedia* and *JRC-Acquis* collections respectively. The selected texts from both collections were split into a testing set, which contained 10.000 documents and a training set for the retrieval model, which contained the remaining texts [12].

In the aforementioned evaluation, the three approaches were compared in a ranking task with the help of three different experiments, which are performed on two different test collections for each model. The languages which were paired with English are Spanish, French, German, Dutch and Polish. For these experiments, we consider the query document d_1, from a test collection D_1, alongside the collection D_2, which contains the documents aligned with the ones in D_1. In addition to this, we also consider d_2, a document aligned with d_1. The three experiments are performed on 1000 randomly chosen texts d_1, with the help of the three approaches **CL-C3G**, **CL-ESA** and **CL-ASA**.

The first experiment is the *Cross-Language Ranking* in which all the documents in the collection D_2 are ranked with respect to their cross-language textual similarity to d_1. The retrieval rank of d_2 is also recorded and it should be on the first or on one of the top ranks in order to be decided if d_2 is a translation of d_1 or not.

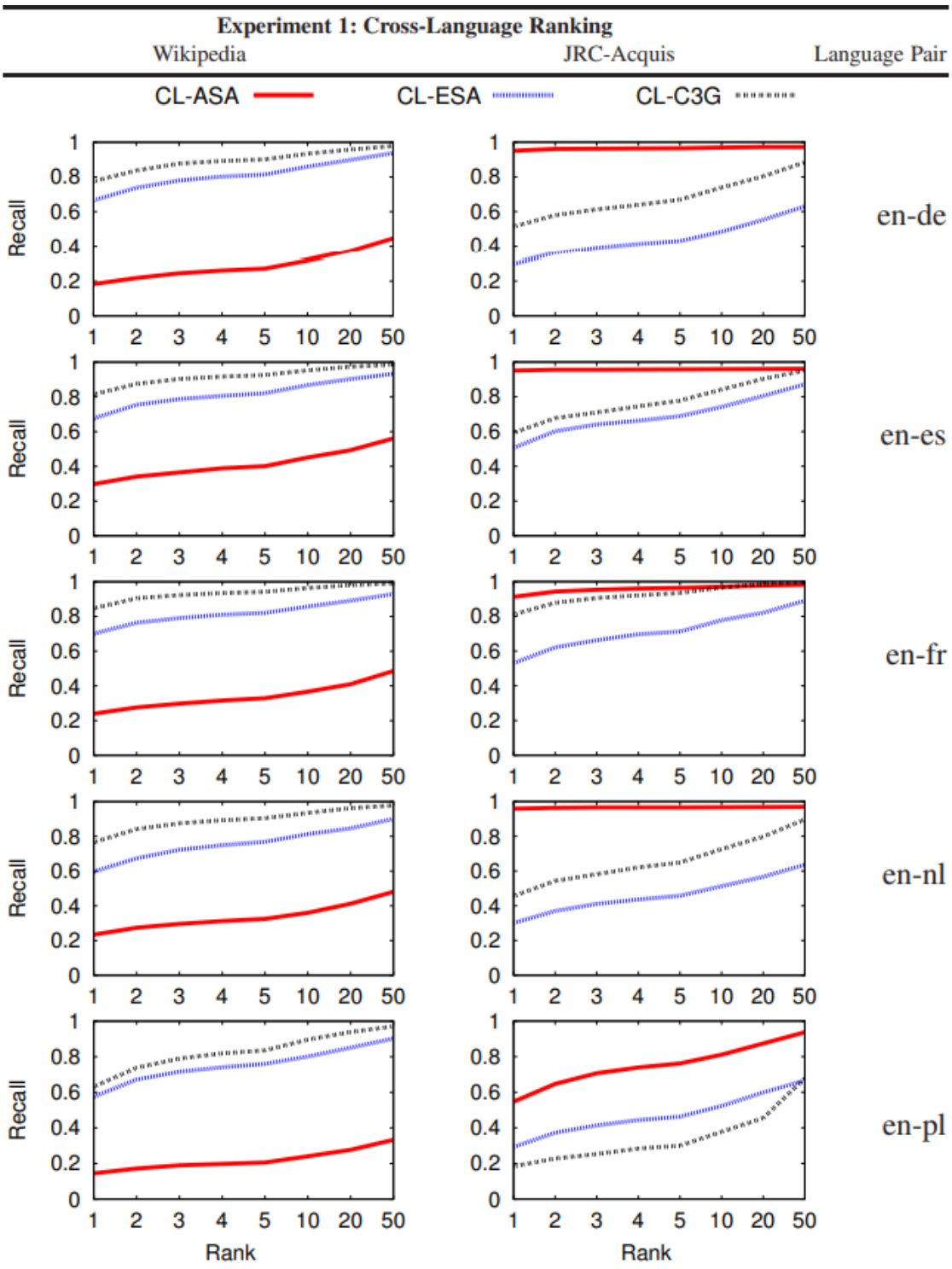

Figure 2.5: Martin Potthast et al. (2010), Results of Experiment 1 for the cross-language retrieval models [12]

The graphic extracted from [12] presents the results of the experiment through plots of recall-over-rank on the different language pairs. It can be observed that unlike **CL-ASA**, which varies in its performance, the other two approaches achieve stable results on both document collections. This can be explained by the fact that *JRC-Acquis* is a collection of parallel documents and **CL-ASA** has proved to work better on translations rather than on comparable corpus. One of the reasons why **CL-ESA** and **CL-C3G** achieve better results on the Wikipedia collection than on the *JRC-Acquis* one might be that the latter might be biased to some extent, since it only contains legislative texts from the European Union and is therefore homogeneous. Due to this, **CL-ASA** appears to be more susceptible to bias than the other two approaches, but it may perform better when a more diverse parallel document collection is used for training [12].

The second experiment is the *Bilingual Rank Correlation* in which given a pair of aligned documents d_1 and d_2 from the collections D_1 and D_2, the texts from the latter are ranked twice. Firstly, they are ranked with respect to their cross-language similarity to d_1, using one of the three approaches and afterwards, with respect to their monolingual similarity to d_2, with the help of the *vector space model*. The **VSM** is an algebraic model used for representing text documents as vectors of identifiers which helps identify whether or not two texts have a similar meaning even when they don't share the same words [15]. After completing the two rankings, the top 100 of both are compared using *Spearman's rank correlation coefficient* ρ which estimates the positive or negative association of the rankings and returns a value in the interval $[-1, 1]$ [18].

Language Pair	**Experiment 2: Bilingual Rank Correlation**					
	Wikipedia			JRC-Acquis		
	CL-ASA	CL-ESA	CL-C3G	CL-ASA	CL-ESA	CL-C3G
en-de	0.14	0.58	0.37	0.47	0.31	0.28
en-es	0.18	0.17	0.10	0.66	0.51	0.42
en-fr	0.16	0.29	0.20	0.38	0.54	0.55
en-nl	0.14	0.17	0.11	0.58	0.33	0.31
en-pl	0.11	0.40	0.22	0.15	0.35	0.15

Figure 2.6: Martin Potthast et al. (2010), Results of Experiment 2 for the cross-language retrieval models [12]

The table extracted from [12] presents the results of the experiment by presenting the ρ values for the different language pairs and document collections. It can be observed that **CL-ASA** performs significantly better on the *JRC-Acquis* corpora than on the *Wikipedia* one. Different from the first experiment, **CL-ESA** has a similar performance to **CL-C3G** and **CL-ASA** on *JRC-Acquis*, however, it continues to outperform **CL-ASA** on the Wikipedia corpus. In addition to this, **CL-C3G** is also outperformed by **CL-ESA** on both document collections. All three models present poor performance on at least one language pairing, but, **CL-ESA** appears to be the most reliable

as a general purpose retrieval approach, while more care needs to be taken when selecting the language pairings used for the other two models [12].

The third and final experiment is the*Cross-Language Similarity Distribution* which provides an indication of what can be expected from each of the three retrieval models. Therefore, the experiment does not directly compare the models, but provides information about the range of cross-language textual similarity values measured when using one of the approaches and in particular, which values relate to a lower similarity and which values relate to a higher similarity.

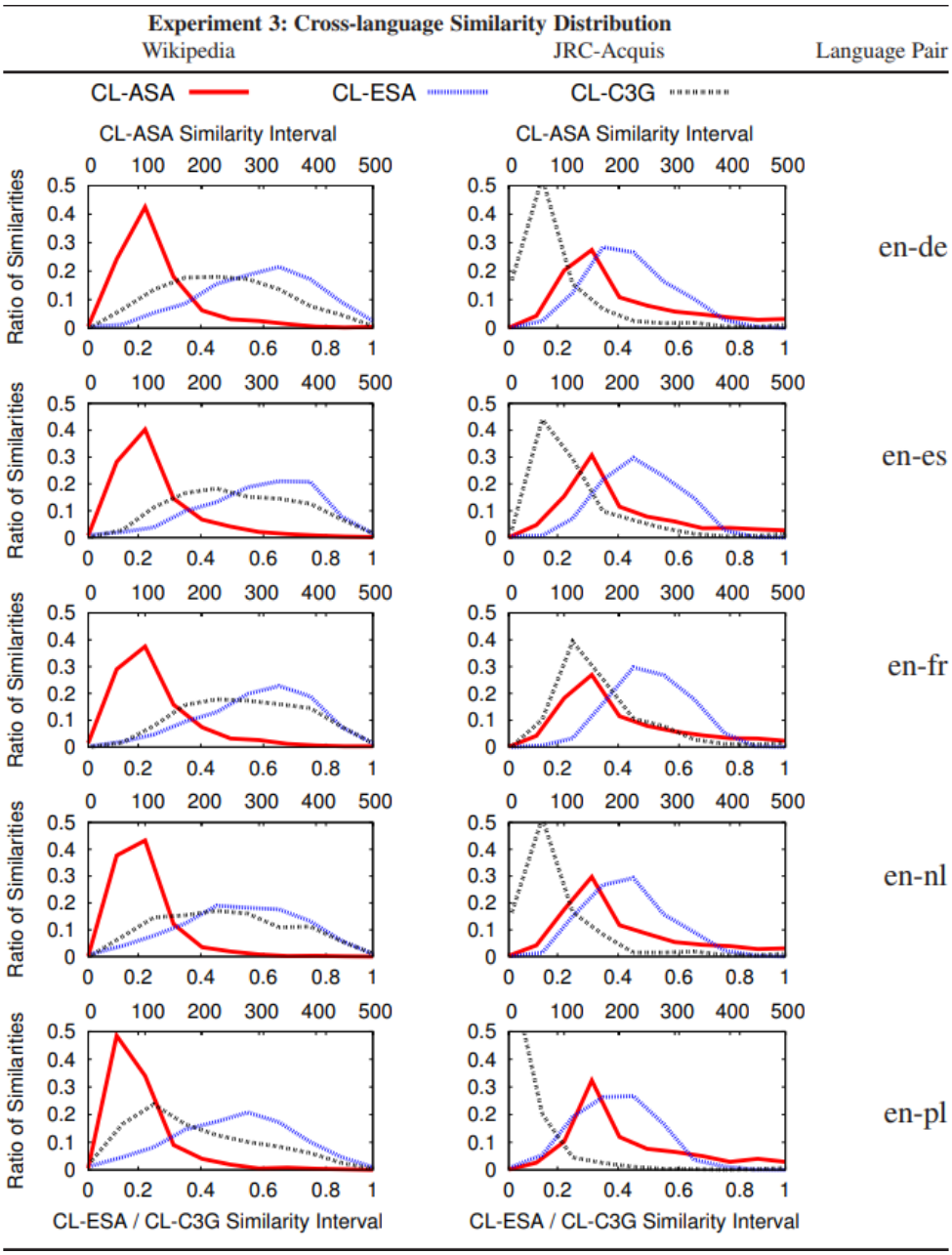

Figure 2.7: Martin Potthast et al. (2010), Results of Experiment 3 for the cross-language retrieval models [12]

The graphic extracted from [12] presents the results of the experiment through plots of similarity-over-similarities on the different language pairs. Since the textual similarities computed by **CL-ASA** are not normalized like for the other two approaches, the similarity distributions has been plotted on a different scale: the top x-axis of the plots presents the range of similarities measured with CL-ASA, while on the opposite side there are the similarities measured with the other models. As a result, the absolute values of the three approaches are not significant, but the order of the documents that they induce is. Because of this, the comparison of the similarity values produced by two different models is not enough to determine which of them performs better. For example, comparing **CL-ESA** with **CL-C3G** by their similarity distribution will result in the former being better, as it is more to the right. However, if looking at the first experiment, the latter outperforms **CL-ESA** [12].

The conclusion of these experiments is that each of the three models has its strengths and weaknesses and therefore one must be careful when choosing which approach is better in which context.

Chapter 3

Project Plan

This section will outline the general plan of the project and will include details about what the end goal will, what extra work can be done if the time permits, what are the key milestones and what is the current status of the work.

3.1 The End Product

The main goal of the project is to produce a working algorithm that can effectively and efficiently determine whether or not a piece of text is automatically translated or not. Since this algorithm has to be showcased and available to the general public, the secondary goal of the project will be the creation of a website that is easy to use, such that any person can benefit from it without too many complications. The website will be done in **React**, using **Typescript**, due to convenience, since this was the technology used during the Software Engineering Group Project. It will follow the KISS principle (keep it "stupid simple"), so it can be easily used by as many people as possible. Throughout the development, people will be asked to test the website and give feedback, which will guide the evolution and shape it into an application that users enjoy interacting with.

3.2 The Documentation

As the most important part of the **Individual Project** is the **Final Report**, all the key aspects will need to be carefully documented during the development of the aforementioned algorithm and website, to produce a significant and useful piece of academic writing. As a result, the final report will not only present and describe the final product, but also the intermediate steps and the obstacles encountered on the way. There is a considerable chance that the algorithm will not function properly on the first or even the second attempt, however, these tries will also need to be documented, for the people that read the paper can get an idea of the bigger picture of what has gone into the development of this detection of automatic translation mechanism.

3.3 The Milestones

- The first milestone will be to have a website with minimal functionality, such that users can try it and offer feedback - 12th of March
- The second milestone will be to develop a detection of automatically translated documents algorithm with a low accuracy - 19th of March
- The third milestone will be to write about the first two milestones in the final report. - 26th of March
- The fourth milestone will be to incorporate the algorithm into the website - 16th of April
- The fifth milestone will be to improve the accuracy of the algorithm. - 3rd of May
- The sixth milestone will be to finalise the website after the algorithm reaches a satisfactory state in terms of the accuracy and recall metrics - 10th of May
- The seventh milestone will be to evaluate the algorithm - 17th of May
- The eighth milestone, if time permits, will be to explore any possible extension. - 31st of May
- The ninth milestone will be to finalise the project report - 10th of June
- The last milestone will be to finalise the project presentation - 18th of June

3.4 The Possible Extensions

One possible extension that would be interesting to look into is instead of only answering the question of whether or not the text is automatically translated or not, to also mention which language was the text most probably translated from. This would probably result in the change of the architecture of the algorithm since this would imply solving a *multi-class classification task*, rather than a *binary classification task*. Therefore, a new model would need to be created, validated and tested, which would require considerable time, which might not be available during the later stages of the project.

3.5 The Current Status

Work has been currently started on the website since I am already familiar with **React** and **Typescript**. The work on the algorithm has not started yet, since I want to research more into *Machine Learning* and develop my **Python** skills further. Therefore, before starting working on the algorithm I want to fully understand how libraries such as *PyTorch* and *TensorFlow* work and I also want to practice more with programming in **Python**.

Chapter 4

Evaluation Plan

This section will outline the key conditions that need to be met for the project to be successful. Since the project has two big parts, both will need to be evaluated. Due to the fact that these two components are different, one being a website and the other being a *Machine Learning* algorithm, different benchmarks will be used.

4.1 The Website

For the website to be considered successful, it will need to be able to showcase the detection algorithm that will be developed. The users will have to be able to easily interact with it and not encounter any unexpected issues. In addition to this, the site will need to be able to handle simultaneous requests, as several users might make use of it at an arbitrary moment in time. The last matter that needs to be addressed is how all the components will scale inside the website, to facilitate use on both desktops and mobile device screens of different resolutions. Testing will be the key to ensuring the correct functioning of the website and it will be split into two essential parts: unit and user testing.

4.1.1 Unit Testing

Due to the use of **React**, unit testing is a perfect fit for ensuring the robustness of the website. Since all the elements, including the pages, take the form of *React Components*, it is convenient to use this form of testing. By checking the functionality of all the *Components*, it can be ensured that up to 100% of the codebase behaves in the way it should.

4.1.2 User Testing

Perhaps the most important part of the website evaluation will be constituted by the interaction of real users with it. Individuals will be requested to navigate through the app and propose suggestions of what can be changed or improved. Since users will be the ones interacting and using the website in their daily lives, the design of it will need to be user-centred. Therefore, instead of only evaluating the final product,

people will constantly evaluate along the way. Thus, the addition of new features will be guided by the received feedback. Due to the fact that users will shape the app into what it will be in the end, enough information will be gathered to prove qualitative aspects such as ease of use, as the website will become more user-friendly after each iteration.

4.2 The Detection Algorithm

This part of the project will use a completely different benchmark for evaluation since the users will not directly interact with it. The algorithm will measure its success through its accuracy. Hence, the higher the accuracy of the final model, the higher the success. It was previously established that the mechanism used for the task at hand will be a *Neural Network* that solves a classification task. Since this is the case, the accuracy will be calculated by taking the number of correct predictions and dividing it over the total number of data points used for testing. However, this will not be the only criteria taken into account when deciding the effectiveness of the algorithm.

4.2.1 Recall

The *Recall* of the algorithm is calculated by taking the number of correctly classified positive examples and dividing it by the total number of positive examples. Therefore, *Recall* is the probability of an example being positive given that it was correctly classified. As a result, the higher this metric, the higher the confidence that the class is correctly recognized.

4.2.2 Precision

The *Precision* of the algorithm is calculated by taking the number of correctly classified positive examples and dividing it by the total number of predicted positive examples. In other words, *Precision* is the probability of an example being classified as positive given that the example is, in fact, positive. Therefore, the higher this metric, the higher the confidence that an example labelled by the *Neural Network* as positive is indeed positive.

4.2.3 Conclusion

After looking into the two aforementioned metrics, it becomes apparent that both precision and recall alongside accuracy are important for the evaluation of the algorithm. However, it is preferable for this specific task to focus on maximizing precision. This is due to the fact that having false positives is worse than having false negatives, in this case, since it is not desirable to accuse a person of cheating when they, did not cheat.

Chapter 5

Ethical Considerations

Since the website that will be developed during this individual project will be part of the World Wide Web, humans will interact with it. However, this does not pose any problems from an ethical point of view, the only way the users will engage with the website will be by submitting a piece of text or a document and getting some sort of feedback. This means that no account creation is required and therefore no personal data of the users will be collected or processed. In addition to this, the texts that the people want to get checked for plagiarism will not be stored for further use after processing, since there are enough publicly available data collections that can be used for the training of the algorithm. Therefore, no major ethical issues can arise regarding the human use or protection of personal data.

The only ethical problems that need to be considered are the legal issues, which will emerge from the use of software with copyright licenses. During the development of the textual similarity detection algorithm, various libraries will be used which will need to be credited accordingly. This will be done on the website, since that is the part of the project that is available to the public. Furthermore, the data sets used for the training of the detection algorithm will need to be credited as well.

In conclusion, although the project does not present many ethical issues, those that will arise will need to be treated carefully and will most certainly involve giving credit where credit is due. In addition to this, a final issue that must be considered is what happens when the algorithm gets the wrong answer. However, this is more of a moral issue rather than an ethical issue.

Bibliography

[1] Jeremy Ferrero et al. *Deep Investigation of Cross-Language Plagiarism Detection Methods*. 2017. URL: arXiv:1705.08828. (accessed: 19.01.2021).

[2] Ralf Steinberger et al. "JRC EuroVoc Indexer JEX - A freely available multi-label categorisation tool". In: (2013). URL: https://arxiv.org/ftp/arxiv/papers/1309/1309.5223.pdf. (accessed: 20.01.2021).

[3] Camelia Ignat Bruno Pouliquen Ralf Steinberger. "Automatic Annotation of Multilingual Text Collections with a Conceptual Thesaurus". In: (2003). URL: https://www.academia.edu/13517905/Automatic_annotation_of_multilingual_text_collections_with_a_conceptual_thesaurus. (accessed: 23.01.2021).

[4] Luis Albert Barron Cedeno. "On the Mono- and Cross-Language Detection of Text Re-Use and Plagiarism". In: (2012). URL: https://riunet.upv.es/bitstream/handle/10251/16012/tesisUPV3833.pdf?sequence=1.. (accessed: 29.01.2021).

[5] Vera Danilova. "Cross-Language Plagiarism Detection Methods". In: (2013), pp. 51–57. URL: https://www.aclweb.org/anthology/R13-2008.pdf. (accessed: 20.01.2021).

[6] McNamee P.; Mayfield J. "Character N-Gram Tokenization for European Language Text Retrieval. Information Retrieval 7". In: (2004), pp. 73–97. DOI: https://doi.org/10.1023/B:INRT.0000009441.78971.be. (accessed: 22.01.2021).

[7] KantanMT. *Machine Translation*. URL: https://kantanmt.com/documents/Machine_Translation.pdf. (accessed: 11.01.2021).

[8] Maik Anderka Martin Potthast Benno Stein. "A Wikipedia-Based Multilingual Retrieval Model". In: (2008). URL: https://webis.de/downloads/publications/papers/stein_2008b.pdf. (accessed: 24.01.2021).

[9] Izet Masic. "Plagiarism in Scientific Publishing". In: (2012). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3558294/. (accessed: 03.02.2021).

[10] Robert C. Moore. "Improving IBM Word-Alignment Model 1". In: (2004). URL: https://www.aclweb.org/anthology/P04-1066.pdf. (accessed: 20.01.2021).

[11] Gupta P.; Barrón Cedeño LA.; Rosso P. “Cross-language high similarity search using a conceptual thesaurus. En Information Access Evaluation. Multilinguality, Multimodality, and Visual Analytics.” In: (2012), 7488:67–75. DOI: doi:10.1007/978-3-642-33247-0_8. (accessed: 22.01.2021).

[12] Potthast M.; Barrón Cedeño LA.; Stein B.; Rosso P. “Cross-Language Plagiarism Detection. Language Resources and Evaluation”. In: (2011), 45(1):45–62. DOI: doi:10.1007/s10579-009-9114-z. (accessed: 22.01.2021).

[13] Ignat C. Pouliquen B. Steinberger R. “Automatic Linking of Similar Texts Across Languages. In: Recent Advances in Natural Language Processing III”. In: (2003). URL: https://books.google.ro/books?hl=ro&lr=&id=hZs6AAAAQBAJ&oi=fnd&pg=PA307&dq=Automatic+Linking+of+Similar+Texts+Across+Languages.+In:+Recent+Advances+in+Natural+Language+Processing+III&ots=BVHy1pblOy&sig=23Qd230mZCrkWH-X7r2jLOxqubE&redir_esc=y#v=onepage&q&f=false. (accessed: 23.01.2021).

[14] Selva Prabhakaran. *Cosine Similarity – Understanding the math and how it works (with python codes)*. URL: https://www.machinelearningplus.com/nlp/cosine-similarity/. (accessed: 23.01.2021).

[15] Kurtis Pykes. *Vector Space Models*. 2020. URL: https://towardsdatascience.com/vector-space-models-48b42a15d86d. (accessed: 02.02.2021).

[16] Melissa S.Anderson. “The problem of plagiarism”. In: (2011). URL: https://doi.org/10.1016/j.urolonc.2010.09.013. (accessed: 03.02.2021).

[17] Chow Kok Kent; Naomie Salim. “Web Based Cross Language Plagiarism Detection”. In: (2009). URL: https://arxiv.org/ftp/arxiv/papers/0912/0912.3959.pdf. (accessed: 30.01.2021).

[18] *Spearman correlation coefficient: Definition, Formula and Calculation with Example*. URL: https://www.questionpro.com/blog/spearmans-rank-coefficient-of-correlation/. (accessed: 03.02.2021).

[19] Bruno Stecanella. *What is TF-IDF?* 2019. URL: https://monkeylearn.com/blog/what-is-tf-idf/. (accessed: 23.01.2021).

[20] Ralf Steinberger; Bruno Pouliquen; Anna Widiger; Camelia Ignat; Tomaž Erjavec; Dan Tufiş; Dániel Varga. “The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages”. In: (2006). URL: https://arxiv.org/ftp/cs/papers/0609/0609058.pdf. (accessed: 02.02.2021).