



Universidad de Alcalá

Grado Universitario en Lenguas Modernas y Traducción
curso académico (2017-2018)

Trabajo Fin de Grado

“Comparación de la adecuación de
las traducciones ofrecidas on-line en
textos de diversos ámbitos”

Clara Nieto Camacho

Tutor

Esperanza Cerdá Redondo

Lugar y fecha de presentación prevista

Universidad de Alcalá (UAH). Facultad de Filosofía y Letras



[Incluir en el caso del interés de su publicación en el archivo abierto]

Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

ACEPTACIÓN DEL TUTOR DEL TRABAJO FIN DE GRADO

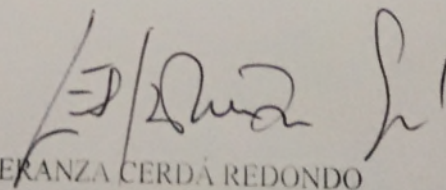
Don/Doña ESPERANZA CERDÁ REDONDO, profesor del Departamento de FILOLOGÍA MODERNA y en su calidad de tutor, **expone** que revisado el Trabajo Fin de Grado "Comparación de adecuación de las traducciones ofrecidas on-line en textos de diversos ámbitos" realizado por Don/Doña CLARA NIETO CAMACHO, estudiante de la Titulación de LENGUAS MODERNAS Y TRADUCCIÓN

Autoriza al estudiante arriba citado a defender su trabajo en la convocatoria de

- Enero
- Julio
- Septiembre X

En Alcalá de Henares, a 16 de SEPTIEMBRE de 2018

VºBº y firma del tutor



Fdo. ESPERANZA CERDÁ REDONDO

A LA COMISIÓN DE TRABAJOS FIN DE GRADO DE LA TITULACIÓN DE



Universidad
de Alcalá

DEPARTAMENTO DE
FILOLOGÍA MODERNA
Colegio San José de Caracciolos
C/. Trinidad, 3
28801 Alcalá de Henares (Madrid)
Telf.: +34 91 885 44 41
Fax: +34 91 885 4445
dpto.filmod@uah.es

DECLARACIÓN DE ORIGINALIDAD DEL TRABAJO DE FIN DE GRADO

D./Doña CLARA NIETO CAMACHO

estudiante del Grado en LENGUAS MODERNAS Y TRADUCCIÓN

DECLARA que presenta su Trabajo Fin de Grado conforme al artículo 2.1 de la Normativa de Trabajos Fin de Grado de la Universidad de Alcalá:

El TFG es una asignatura obligatoria en todos los estudios de Grado. Su contenido consistirá en un trabajo original, autónomo e individual que cada estudiante realizará bajo la orientación de un tutor, que le permitirá mostrar de forma integrada los contenidos formativos recibidos y las competencias adquiridas asociadas al título de Grado. El término original queda referido a que en ningún caso pueda ser un trabajo plagiado ni presentado con anterioridad por el alumno en alguna otra asignatura, no siendo necesario que sea un trabajo inédito.

En ALCALÁ, a 11 de SEPT de 2018.

Fdo.:

COMISIÓN DE TRABAJOS FIN DE GRADO DE LA TITULACIÓN DE LENGUAS MODERNAS

RESUMEN

En el mundo de hoy en día existe una cantidad creciente de traducciones generadas de manera automática por herramientas de traducción. Ante este hecho, merece la pena preguntarse por la validez de estos métodos y si las traducciones que generan son de calidad suficiente para no precisar de la revisión posterior por un traductor. En este Trabajo Fin de Grado se busca evaluar un par de herramientas de traducción automática (Google Translate y OpenNMT) con el uso de los estimadores de calidad más habituales (BLEU, TER, NIST, METEOR, CHRF) en diversos ámbitos lingüísticos (textos de origen jurídico, comercial, literario, foros online y subtítulos de películas). Para ello se toman entre 300 y 500 líneas de varios textos en inglés, y se buscan o se elaboran traducciones consideradas correctas, que se emplearán como traducciones de referencia. Obtendremos igualmente traducciones automáticas, a las que denominaremos hipótesis o candidatas. Tras un formateado y preprocesado, dichos textos serán sometidos a un programa encargado de realizar sucesivamente el cálculo de los estimadores. Se representan los resultados en forma de gráfica y se sacan conclusiones. De esta forma se considerará qué textos, debido a sus características intrínsecas, son más adecuados para ser traducidos automáticamente y cuáles presentan más dificultades y requieren mayor trabajo de post-edición.

PALABRAS CLAVE

Traducción automática; calidad; comparación de métricas; estimadores de adecuación; lenguajes de programación; ámbitos lingüísticos; preprocesado; estadísticas; n-gram.

ABSTRACT

Nowadays there are a growing number of translations which are product not of a human translator, but of a machine. An enormous quantity of words in a variety of languages is spewed from automata when fed with text in a source language. In terms of items translated, the productivity is fabulous, but what about the quality of the generated translations? As anyone can corroborate, in a great many occasions they are in dire need of further human-supervised translation. In this end-of-degree project we aim to evaluate a couple of Machine Translation (MT) tools (GoogleTranslate and OpenNMT) using standard adequacy estimates (BLEU, TER, NIST, METEOR, CHRF) translating texts taken from different linguistic fields (legal, commercial, literary, online forums and film captions). For this purpose, we have extracted around 300 to 500 lines from English source texts considered representative of those categories. When possible, we took official reference translations, but when there was no reference text, we made our own translation. In a similar fashion, we used the aforementioned MTs to obtain the candidate translations whose adequacy will be quantified. After formatting and pre-processing, these texts will be used as input to a program, which will calculate the desired estimators. Numeric results are presented as a bar graph to draw conclusions more easily. Therefore, this will show which texts are more suitable to be translated with MTs, due to their intrinsic features, and which ones prove to be more problematic and require more post-edition effort.

KEYWORDS

Machine translation; quality; comparison of metrics; adequacy estimates; programming languages; linguistic fields; pre-processing; statistics; n-gram.

DEDICATION

I would want to express my gratitude to everyone who has been by my side and helped me during the development of this project, especially to my parents Marcos and Sagrario. Without their deep support this work wouldn't have been possible, and therefore this end-of-degree project is dedicated to them.

TABLE OF CONTENTS

TABLE OF CONTENTS.....	10
1 INTRODUCTION	12
1.1 Reason for the present work.....	12
1.2 Objectives.....	12
2 STATE OF THE ART	12
3 DISSERTATION	14
3.1 Requirements.....	14
3.1.1 Information Technology.....	14
3.1.2 Origin of the employed texts.....	15
3.2 Evaluation metrics.....	18
3.2.1 BLEU (BiLingual Evaluation Understudy).....	18
3.2.2 TER.....	20
3.2.3 NIST	20
3.2.4 METEOR (Metric for Evaluation of Translation with Explicit ORdering)	21
3.2.5 CHRF (Character n-gram F-score).....	21
3.3 Implementations of the different scores used.....	21
3.4 Description of the method.....	22
3.4.1 Origin and character of the data	22
3.4.2 Texts chosen.....	23
3.4.3 File text formatting.....	27
3.4.4 Obtaining texts to be used as translation references and candidates.....	29
4 RESULTS.....	31
5 CONCLUSIONS	36
6 REFERENCES	37
7 ANNEXES	39
7.1 Figures	39
7.2 Code	44
7.2.1 procesa-legal.sh.....	44
7.2.2 lee-comentarios.py	45
7.2.3 busca-multiples-claves.py	46
7.2.4 bleu-ss.py	47
7.3 Zusammenfassung.....	50

1 INTRODUCTION

1.1 Reason for the present work

Submitted to the Department of Modern Philology in partial fulfillment of the requirements for the Degree of Bachelor of Translation Studies.

1.2 Objectives

In today's world the commercial and political interests to acquire global audiences have prompted an enormous growth of technologies making possible obtaining basically correct translations between languages, in a field called Machine Translation (MT). As the volume of translated texts increases exponentially, an urgent need arises to assess the quality of the obtained translations, using different approaches. At first these were obtained by professional translators that checked the quality of the MT output; today this is no longer a must, and sometimes evaluations that are also automatic are needed. These metrics try to compare themselves with human judgement (e.g. if the meaning and the form of the original texts are adequately conveyed in translations). In this paper we will make use of some of the most classical metrics and other recently proposed, as information that could be useful to translators who make use of today's machine translation applications.

Due to the method we have chosen to follow, to the small sampling of the various texts, and to the fact that many of the steps are performed manually, this study does not allow us, nor pretends to be statistically significant. Therefore, the conclusions we have drawn are exclusively circumscribed to the data with which we have worked.

2 STATE OF THE ART

Machine Translation (MT) arises from pioneering work in IBM in the decade of 1960. But it was one by-product of the cold-war effort by the U.S. Armed Forces to be informed on the continuous scientific advances of the U.S.S.R., especially in the fields of physics and mathematics. Later on, similar systems were developed by the British to follow from Hong Kong developments in Communist China. The computing resources were very scarce by today's standards and also the conceptual models. In contrast, the objective was too ambitiously stated: "Fully Automatic High-quality Machine Translation". This led to a very critical review in 1966 by the National Academy of Sciences, which suggested abandoning the effort entirely.

After that initial disappointment, a change of objective took place: the systems developed should have a quality good enough to just require minimal human revision, with the possibility of obtaining economic benefits commensurable with their cost.

The SMT (Statistical Machine Translation) approach is to build a logic machine that learns the relations between two languages, by establishing a probabilistic function from one sentence in the source language to many possible sentences in the target language, each of them with their own probability.

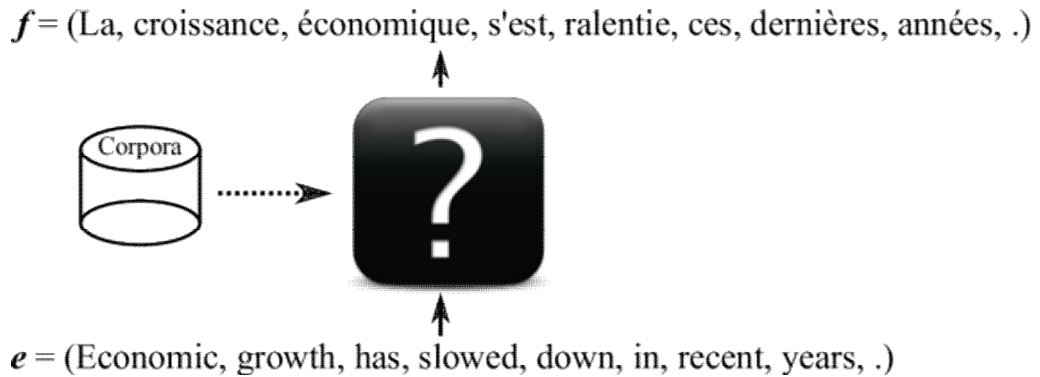


Illustration #1. The concept behind MT, when from a source language we want to obtain a translation in another language, based on the information provided by corpora¹

The statistical models consist of translation and language models. The translation model represents the probable word translations and is trained by bilingual data, which consist of sentence pairs in two different languages. The language model encodes the sentence fluency and is trained by the target language data. The decoder searches for the most likely target word sequence from a large number of hypotheses using these two models. SMT enables us to construct robust translation systems with low cost in short development cycles if the training data is available. SMT is especially well suited when the word order between the source and target languages is similar. The translation tables can be domain-specific, that is, they adopt the style and vocabulary of the language used for training; in this way we can build a MT machine specifically tuned for translating legal text. Obtaining proper parallel corpora of bilingual texts is critical.

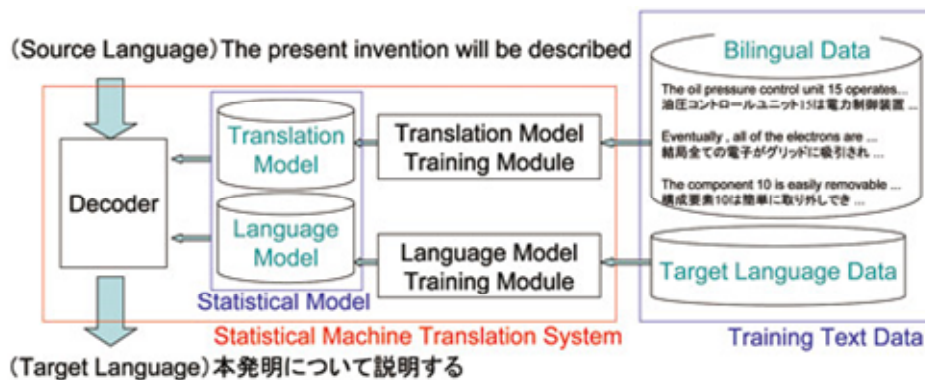


Illustration # 2. Conceptual model of SMT²

¹ Taken from Kyunghyun Cho “Introduction to Neural Machine Translation using GPUs”, <https://devblogs.nvidia.com/introduction-neural-machine-translation-with-gpus/> [accessed 05/19/2018]

² Illustration taken from the web page of NTT Communication Science Laboratories (http://www.kecl.ntt.co.jp/rps/english/Research_e/cn20/research_innovative03_e.html) [accessed on 05/19/2018]

With the irruption of neural net technology SMT has evolved, either adopting it as a new “module” or as the “kernel” of the system. Illustration 3 shows some of the approaches for the adoption of this technology. A “pure neural network system” is referred as NMT (Neural Machine Translation).

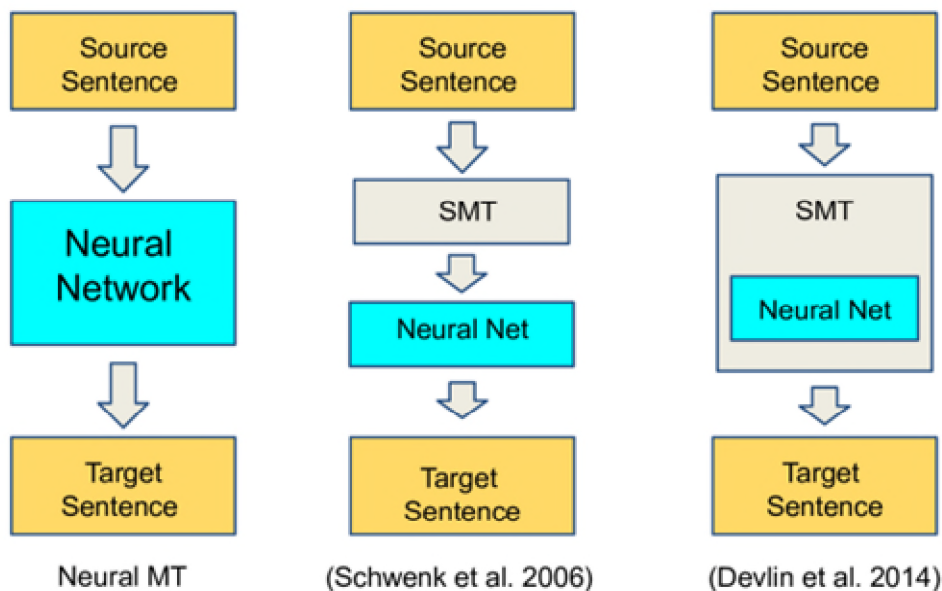


Illustration # 3. Neural MT, SMT+Reranking-by-NN and SMT-NN. Based on a presentation of Bahadanau et al (Neural Machine Translation by Jointly Learning to Align and Translate, 2014).

Today there is an ever-increasing competition to find the best combination for a particular job. Progress in NMT has been impressive in the last years, but it still hasn't a definitive advantage over SMT (Castilho, S. et al, 2017).

In this paper we have employed the web services **Google Translate** and **OpenNMT**, both of them employing NMT in a varying degree. Figure 1 in the Annexes illustrates this process. We tried to also use a “pure SMT system” as **MOSES** for comparison purposes, but it proved beyond our computing and knowledge possibilities, so we won't be able to enter in this particular discussion on which system is better nowadays.

3 DISSERTATION

3.1 Requirements

3.1.1 Information Technology

At first, we tried to install a SMT system such as MOSES in order to learn about it. The hardware and software requirements were relatively high, but we finally settled for a system barely able to compute the adequacy metrics all the previous effort found some justification. While trying to reproduce an SMT system, we installed all the basic software needed to run the MT toolkits.

The models for obtaining the adequacy metrics are widely available and continuously “mutating”, that is, the scientific community greatly encourages innovation

and experimentation in this field and finds in an open-source approach the ideal environment for this. This is the reason why the systems developed are very modular, because such structure permits small changes, in contrast with large, monolithic systems. Usually they are comprised of a series of programs, that run sequentially each one using the input from the preceding module and generating output that will in turn serve as input for the next module. The programs will consist either in machine-code executables or by scripts in interpretable languages. In our case, we only have made use of the latter.

The interpreted programming languages installed were the following: Python version 3.5 or later; Perl, Ruby and Java³. The scripts that direct the execution of the modules are written as command interpreter batch programs, based exclusively on the Unix command interpreter in its bash version (Bourne Shell) of Linux. A 32-bit machine with 2 Gb of RAM running Linux Ubuntu 16.04 proved more than enough for our purposes, offering an almost assured compatibility between modules. Following abundant advice available on internet, we installed specialized utilities such as Anaconda⁴, Cpan⁵, Pip⁶, Git⁷ and others in order to gain easy access to the different software libraries that the modules make use of or the source code of applications. The character set of choice was UTF-8 to assure compatibility.

3.1.2 Origin of the employed texts

In the limited scope of this paper we have chosen texts from five distinct linguistic domains.

3.1.2.1 *Legal Texts*

Corpora of bilingual texts are relatively easy to obtain, especially those of the European Union, whose legal texts are translated to a wide variety of official languages of the EU. For obvious reasons, the translations have a very good level and the style and vocabulary is standardized. In our case we chose a legal text that had official English and Spanish versions.

- Consolidated Version of The Treaty on the Functioning of the European Union (Official Journal of the European Union, 7/06/2016)

³ The criteria for installing new software was simple: when an existing script complains something is missing, try to locate and install it. It could be described as trial and error and it's not so easy as it seems.

⁴ Anaconda. Python Data Science Platform. Copyright 2018 Anaconda, Inc. All Rights Reserved.

⁵ Comprehensive Perl Archive Network (www.cpan.org)

⁶ Python Packaging Authority (www.pypa.io)

⁷ git-scm.com

- Versión Consolidada del Tratado de Funcionamiento de la Unión Europea (Diario Oficial de la Unión Europea, 7/06/2016)

These texts are similar in character to the ones employed in MT for training⁸ and prove themselves to be particularly optimized, as our results will show.

3.1.2.2 *Informal Texts (Colloquial Writing)*

Spoken language is inherently varied and problematic. As a very general approximation to it, we have chosen the commentaries submitted to an Amazon Product Forum: Videogames. The commentaries are intermediate between common talk and specialized jargon.

The data as such is readily available⁹ for the period May-96 to July-14. Of it, we have chosen what Julian McAuley names as “small experiment subsets” in the Section called *User Reviews (Video Games Section)*. In our case, translations of them aren’t available, so we will use an alternate MT engine such as Bing Microsoft Translator to obtain the reference translation, and afterwards we will check thoroughly the output.

3.1.2.3 *Commercial Texts*

Distance selling is a commercial relationship where a consumer and a supplier running an organised distance-selling scheme do not meet face to face at any stage until after the contract has been concluded. The means employed today to put into communication both parts include the traditional press adverts accompanied by order forms, catalogues and telephone calls. However, the force driving today this business have an increasingly complex information infrastructure, ranging from simple tele-shopping, to mobile phone commerce (m-commerce) and the use of the Internet (e-commerce).

From humble origins like a bookseller, companies like Amazon have used the new communication possibilities to acquire a global clientele, through what is called an “Internet Portal”. The effort behind such a push has been immense, including complex logistic capabilities and the development of on-line distributed systems where people can search goods in text form, order them and have them delivered to their homes or business. The relationship between the would-be-purchaser and the sales portal has very limited interactivity, so the customer can ask questions to the seller and read opinions about the products before reaching a decision. And as the business expands, more and more audiences and languages are thrown into the mix, so a comment made in Finnish is made available in Spanish to speakers of this last language, for example. This is not the product of an army of on-line translators, but of MT combined with Artificial Intelligence (AI) tools.

⁸ Specifically, the EUROPAL corpus.

⁹ McAuley, Julian. *Amazon product data*. (<http://jmcauley.ucsd.edu/data/amazon/>). File *reviews_Video_Games_5.json.gz* [retrieved on 04/13/2018]

Following the lead of Amazon, other business models have surged, as eBay, in itself a mix of a simple fixed price on-line catalogue with something like an auction room. In their portal, business owners and mere users can offer their products, ranging from mass-produced to vintage or antique objects. The structure of the advert is more or less always the same: a fixed length string 80 characters long that tries to describe succinctly the object advertised, and a comment of variable size, accompanied by images. The on-line catalogue is really a living database, continuously changing and what users do is search it for goods unknowingly using Application User Interfaces (API), navigating through a hierarchical tree of categories and sub-categories.

In order to avoid the tedious and error-prone approach of simply navigating eBay's pages and cut and copy the adverts, we used a Python library with calls that permit to retrieve the information in a structured format directly from the underlying database. Using this means we have obtained those elements marked as "Painting" in eBay's UK portal. As with Amazon comments, no Spanish version of these adverts exists, so we made our own translation of them to use it as reference.

3.1.2.4 *Literary texts*

When dealing with literary texts, we must take into account that the choice of words and style are not arbitrary. On the contrary, this kind of texts are artistic in a sense. The structure of the text and how the message is conveyed has equal importance as to the content itself. A human translator should have a skilful usage of words and sense of beauty to reproduce the literary piece. Therefore, this genre may be very relevant for analysis since we will evaluate how MTs deal with the form and grace of the message.

In order to not to get lost in the vast ocean of literature, we searched for out-of-copyright literary works. We finally decided to stick to the narrative genre, taking English reference texts, and their translations into Spanish as well from a blog with bilingual pairs. This blog seemed to have educative purposes, so it was very clear and systematically structured. Source texts come from Wikisource and many of them have been revised and aligned by a translator¹⁰.

3.1.2.5 *Subtitles (film captions)*

Keeping in mind how desirable and interesting is to cover as many registers as possible in this dissertation, we decided to study some subtitles of film dialogues. Linguistically, subtitles are very interesting since they cover the wide breadth of genres,

¹⁰ "About Me" (<http://www.farkastranslations.com/>) [visited on 05/15/2018]

from colloquial language or slang to narrative discourse. For our purpose we chose to make use of the former since we already had our narrative genre already covered.

In OpenSubtitles¹¹ we found a vast source of film captions, a big collection of parallel corpora in over 60 languages¹². We decided to make use of the subtitles of the the Korean horror movie “The Loner” (Park Hae-Sik, 2008). The file chosen was **en-es.txt.zip**, which expands into two files: **OpenSubtitles2018.en-es.en** and **OpenSubtitles2018.en-es.es**. Both have the same number of lines (61,434.251 using *wc -l*), so we extracted the first 500 from them with the following commands:

```
$ cat OpenSubtitles2018.en-es.en | head --lines=500 > OpenSubtitles2018.en-es.500.en
```

```
$ cat OpenSubtitles2018.en-es.es | head --lines=500 > OpenSubtitles2018.en-es.500.es
```

3.2 Evaluation metrics

All automatic evaluation methods are based on the idea of comparing candidate translations with their corresponding reference translations. The fact that the relationship between a text and its translations is not biunivocal is what makes everything complicated. Evaluation made by humans have the disadvantages of non-reusability and subjectivity. In contrast, automatic evaluation is reusable, fast and even more important, free (Hadla et al, 2015).

Several methods of automatic evaluation have been proposed, but for the purpose of this study we will limit ourselves to four of the most commonly used: BLEU, NIST, TER and METEOR. There are several implementations of each of them, but for the most part we will employ those distributed with the *Carnegie Mellon Multi-Engine Machine Translation Scheme*¹³

3.2.1 BLEU (BiLingual Evaluation Understudy)

One of the most widely used automated methods for evaluating machine-translation quality is the Bilingual Evaluation Understudy, also known as BLEU. It is based on the premise that the more similar one translation is compared to a reference

¹¹ P. Lison and J. Tiedemann, 2016, OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)

¹² <http://opus.nlpl.eu/OpenSubtitles2018.php>. As P. Lison demands, we refer to www.opensubtitles.org as the source.

¹³ Heafield, K., Lavie, A., “Combining Machine Translation Output with Open Source. The Carnegie Mellon Multi-Engine Machine Translation Scheme”, The Prague Bulletin of Mathematical Linguistics, 93 January 2010, pp. 27–36.

translation, the more quality that translation has.¹⁴ In other words, it evaluates how well the translated text scores against reference texts.

BLEU can be estimated making use of more than one reference translation, but in this work for the sake of convenience we will limit ourselves to one. Either by taking a text as a whole or sentence by sentence, BLEU can be calculated. The goodness of fit between the candidate translations (those obtained using machine translators) and the reference translations (those considered correct) is estimated with n-grams. Put in short, these n-grams are sequences of n items of a text, and the number of elements that form an n-gram is the N-gram order. These n-grams range from single words (1-gram or unigram) to sets of two or more adjacent words (2, 3, 4-gram, etc. See Figure 4 in Annexes).

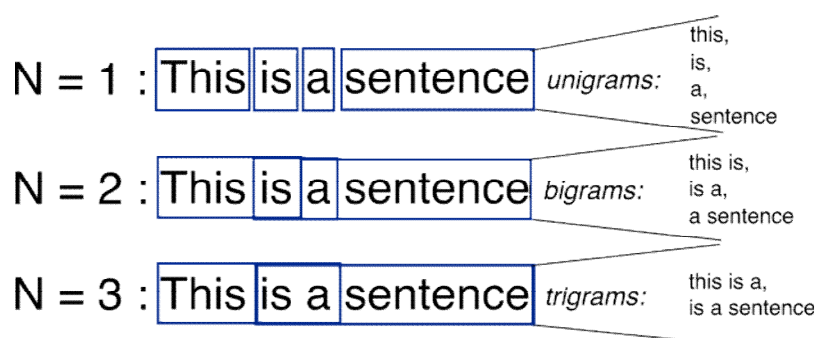


Illustration #4. Visual representation of N-grams of different order.¹⁵

Some of the parameters that control BLEU scores are:

- N-gram order (N). Usually N=4.
- Case sensitivity. For our purposes, we will generally simplify texts to lowercase.
- Brevity penalty (ρ)

Mathematically expressed¹⁶ the way to obtain BLEU scores requires the precision or percentage of n-gram tuples in the candidate (sometimes called *hypothesis*), that occur in the reference(s):

$$P(i) = \frac{Matched(i)}{H(i)}$$

¹⁴ Turian, J. P. et al. 2006. Evaluation of Machine Translation and its Evaluation. New York University Press.

¹⁵ <http://recognize-speech.com/language-model/n-gram-model/comparison> [visited on 05/29/18]

¹⁶ We use Mei-Yuh Hwang's (Washington University) explanation, (<http://ssli.ee.washington.edu/~mhwang/pub/loan/bleu.pdf>).

where $H(i)$ is the number of i -gram tuples in the candidate translation. For a candidate of length n words, $H(1) = n$, $H(2) = n - 1$, $H(3) = n - 2$, etc.

$$Matched(i) = \sum_{t_i} \min\{C_h(t_i), \max_j C_{h_j}(t_i)\}$$

where t_i is an i -gram tuple in hypothesis h ; $C_h(t_i)$ is the number of times t_i occurs in the candidate translation; $C_{h_j}(t_i)$ is the number of times t_i occurs in the reference j ¹⁷.

$$BLEU_a = \left\{ \prod_{i=1}^N P_i \right\}^{1/N}$$

The algorithms penalize short translations since the shorter a sentence is, the more probable a coincidence between candidate and reference translation is. This is called brevity penalty.

$$\rho = \exp\left\{\min\left(0, \frac{n - L}{n}\right)\right\}$$

$$BLEU_b = \rho BLEU_a$$

Where n is the length of the candidate and L is the length of the reference. When there is only one reference, which happens to be our case, L is the number of words in the reference.

BLEU scores a continuous range between 0 and 1, with values closer to 1 being the more similar to the reference text. In order to avoid decimal numbers, sometimes in reference works BLEU and other adequacy estimates are expressed in percentages ($BLEU \text{ score} \cdot 10^2$). Example: 1.423 % instead of 0,01423.

3.2.2 TER

TER (Translation Edit Rate) (Matthew Snover et al, 2006) is a very common metric. TER correlates with the number of necessary changes to convert a translation obtained via machine translation with one used as reference. In its variant called HTER (human), a person is the one who evaluates and commits the required edits. Same as BLEU, its value ranges between 0 and 1.

3.2.3 NIST

NIST is the acronym of the National Institute of Standards and Technology, a normalization institution in the U.S.A. Based on BLEU, it developed its own algorithm which was named N-gram Co-occurrence Scoring (Doddington, G., 2002), but is exclusively known as NIST in the MT world. It is seen as an upgrade to the BLEU metric

¹⁷ As a reminder, in this essay we will only use one reference sentence, so $j=1$.

as it grants more weight to longer n-gram matches than shorter ones. NIST makes available this scoring tool from their web page¹⁸.

3.2.4 METEOR (Metric for Evaluation of Translation with Explicit ORdering)

METEOR (Denkowski, M. et al. 2014) is a very popular metric, second only to BLEU, but much more complex. METEOR has stemming and synonym matching, along with the standard exact word matching. While BLEU and NIST are based on precision alone, METEOR uses precision and recall¹⁹.

3.2.5 CHRF (Character n-gram F-score)

This is a metric which takes into account some morpho-syntactic phenomena, correlates well with human judgment and is language and tokenization independent. It is based in character n-grams when BLEU uses word n-grams. In an additional paper (Popovic, M., 2016), it is stated that the most interesting scores for translations from English sources are CHRF2 and CHRF3, where 2 and 3 stand for the values of the β parameter. Simply speaking, the higher the score, the better.

3.3 Implementations of the different scores used

As explained before, in our work we decided to use directly the adequacy metric programs and pre-processing tools that usually are used to evaluate the output of automatic translation. A script that offers the enormous convenience of running a series of different tests on the same input files is **score.rb**²⁰ (Kenneth Heafield et al, 2010). Such Ruby script is part of the *Carnegie Mellon Multi-Engine Machine Translation Scheme*²¹. It computes sequentially scores such as BLEU, NIST, TER and METEOR²².

The input to this script consists in a translation candidate (hypothesis) whose accuracy we want to measure against one or more reference translations. All files must have been previously pre-processed.

¹⁸ The current version is mteval, version 13A. (<https://www.nist.gov/itl/iad/mig/tools>)

¹⁹ Recall means in the MT context the proportion of the matched n-grams out of the total number of n-grams in the reference translation. It reflects in what degree the translation covers the entire content of the translated sentence (Lavie et al, 2004).

²⁰ <https://github.com/kpu/MEMT/tree/master/Utilities/scoring>

²¹ Heafield, K., Lavie, A., “Combining Machine Translation Output with Open Source. The Carnegie Mellon Multi-Engine Machine Translation Scheme”, The Prague Bulletin of Mathematical Linguistics, 93 January 2010, pp. 27–36.

²² METEOR did not work as expected when it was called from score.rb. We had to resort to execute it from the command-line with all the necessary inputs and options.

An example of the use of **score.rb**:

```
$ ./score.rb --language es --hyp tratado-ue/C1-google-pp.txt --ref tratado-ue/R1-pp.txt --output tratado-ue
```

After a few tries, everything runs smoothly except the METEOR score. Its results were systematically 0, and not a single error message appeared that could help us. As we could not easily fix the **score.rb** script, we found a better approach to look at the temporary files the script generates and which serve as bridges between the applications. With that and after reading the METEOR documentation²³, we settled for the following command line:

```
$ java -Xmx2G -jar meteor-1.4/meteor-1.4.jar tratado-ue/C1-google-pp.txt tratado-ue/R1-pp.txt -l en -norm
```

Once a command line had worked as intended, we composed our scripts in bash, so simple that the only error messages we obtained came from misspelling of the files involved and so were very easy to correct.

3.4 Description of the method

3.4.1 Origin and character of the data

The data employed as sources in this study must meet a series of requirements:

- Must be in pure text form, that is, must only contain alphanumeric and punctuation characters. Absolutely no formatting information.
- The English and Spanish texts used as source and reference must be parallel, that is, sentence-aligned between them.
- The character set employed must include the diacritic and punctuation characters employed in modern Spanish.
- The character set must be platform independent, offering consistently the same appearance between operating systems and text editors.
- The two previous requirements made the use of the UTF-8 character set especially useful.
- If possible, an official translation should be available. As this is only found on legal texts, it was necessary to make use of published translations where available. If not, make our own translation (see Figure 3 in Annexes).
- Except of cases where official translations are readily available, limit ourselves to small corpora, about 500 sentences each. As a trade-off, the statistical significance of the results will be low.

²³ <https://www.cs.cmu.edu/~alavie/METEOR/examples.html>

- To avoid excessive bias, the chosen texts should be representative of a variety of language uses: legal, commercial, spoken. As this is more easily said than done, several compromises have been made.

3.4.2 Texts chosen

Based on the above presumptions, five text sources with very different characteristics have been chosen, each of them trying to resemble closely one of the domains (or literary genres).

- Legal Texts: use of the bilingual text (EN/ES) of the Consolidated Version of the Treaty of the European Union.
- Literary Texts: excerpts from bilingual editions²⁴ of “Voyage to the Centre of the Earth” (Jules Verne), “Ball of Fat” (Guy de Maupassant) and “The Metamorphosis” (Franz Kafka).
- E-commerce Texts: electronically retrieved from eBay.co.uk.
- Informal texts (colloquial writing): in this case the texts chosen are written in Games Forums, a material that closely resembles speech between pc gamers.
- Subtitles (film captions): we took the parallel subtitles (EN-ES) of a Korean horror film from www.opensubtitles.org

3.4.2.1 Legal texts

Original text	Official translation
<p>TITLE XXII TOURISM Article 195 1. The Union shall complement the action of the Member States in the tourism sector, in particular by promoting the competitiveness of Union undertakings in that sector. To that end, Union action shall be aimed at: (a) encouraging the creation of a favourable environment for the development of undertakings in this sector; (b) promoting cooperation between the Member States, particularly by the exchange of good practice.</p>	<p>TÍTULO XXII TURISMO Artículo 195 1. La Unión complementará la acción de los Estados miembros en el sector turístico, en particular promoviendo la competitividad de las empresas de la Unión en este sector. Con este fin, la Unión tendrá por objetivo: a) fomentar la creación de un entorno favorable al desarrollo de las empresas en este sector; b) propiciar la cooperación entre Estados miembros, en particular mediante el intercambio de buenas prácticas.</p>

²⁴ “Learn Spanish with Bilingual Stories” (<http://learn-spanish-with-bilingual-stories.weebly.com/>) [Visited on 05/15/2018]

2. The European Parliament and the Council, acting in accordance with the ordinary legislative procedure, shall establish specific measures to complement actions within the Member States to achieve the objectives referred to in this Article, excluding any harmonisation of the laws and regulations of the Member States.

2. El Parlamento Europeo y el Consejo, con arreglo al procedimiento legislativo ordinario, establecerán las medidas específicas destinadas a complementar las acciones llevadas a cabo en los Estados miembros para conseguir los objetivos mencionados en el presente artículo, con exclusión de toda armonización de las disposiciones legales y reglamentarias de los Estados miembros.

Advantages: Existence of a very important corpus of bilingual texts in the major languages of the European Union legislation. Texts are strongly structured, in a hierarchical fashion. The typography employed usually reflects the category of the following text (title, subtitle...). Conservatism is the norm. Extended use of paragraphs to gather related information is the norm. It employs a limited vocabulary and avoids archaisms. It is especially apt for Machine Translation because all translation engines include these texts in their training.

Disadvantages: Unnatural expressions and vocabulary. Possible bias arising from the fact that MT are usually trained with this kind of corpora.

3.4.2.2 *Literary texts*

Original text	Translation from Wikisource
<p>One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin.</p> <p>He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment.</p> <p>His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked.</p> <p>"What's happened to me?" he thought. It wasn't a dream.</p>	<p>Una mañana, tras un sueño intranquilo, Gregorio Samsa se despertó convertido en un monstruoso insecto.</p> <p>Estaba echado de espaldas sobre un duro caparazón y, al alzar la cabeza, vio su vientre convexo y oscuro, surcado por curvadas callosidades, sobre el cual casi no se aguantaba la colcha, que estaba a punto de escurrirse hasta el suelo.</p> <p>Numerosas patas, penosamente delgadas en comparación al grosor normal de sus piernas, se agitaban sin concierto.</p> <p>—¿Qué me ha ocurrido? No estaba soñando.</p>

Advantages: Almost none. We must remember we use only one reference translation against with the hypothesis is compared.

Disadvantages: These texts exhibit enormous variability, because of the diversity of possible literary genres: biography, narrative, poetry, etc. The richness of the syntax and morphology is a burden for MT. There is an extensive use of punctuation. The goal of the

translator is to remain true to the original and that is very difficult to achieve, especially when the translation is from an intermediate version in a language different from the original. Their authors have also an almost infinite span of writing styles and vocabulary and sometimes they rearrange texts at will²⁵, a big quandary for MT evaluation. In that line, we must punctuate that the process to obtain literary candidate translations proved to be more bothersome than other genres. Some of the tools that we used to get those translations such as OpenNMT were “too intelligent” for our purposes. Sometimes they reformulated sentence structures, altering it and breaking completely the alignment between parallel sentences. This was way too inconvenient, and therefore we decided to simply remove the problematic paragraphs.

3.4.2.3 Informal texts (colloquial writing)

Original text	Our translation
Installing the game was a struggle (because of games for windows live bugs). Some championship races and cars can only be "unlocked" by buying them as an addon to the game.	La instalación del juego fue pesada (debido a los fallos de los juegos de Windows Live). Algunos coches y carreras de campeonato sólo se pueden
I paid nearly 30 dollars when the game was new.	"desbloquear" al comprarlos como un añadido para el juego.
I don't like the idea that I have to keep paying to keep playing.	Pagué casi 30 dólares cuando el juego era nuevo.
I noticed no improvement in the physics or graphics 6 compared to Dirt 2.	No me gusta la idea de tener que seguir pagando para seguir jugando.
I tossed it in the garbage and vowed never to buy another codemasters game.	No noté ninguna mejora en la física o los gráficos comparado con DiRT 2.
I'm really tired of arcade style rally/racing games anyway.	Lo tiré a la basura y juré no comprar otro juego de Codemasters. En cualquier caso estoy realmente cansado de los juegos arcade de estilo rally/carreras.

Advantages: The syntax and the conveyed ideas are very simple. Ample use of buzz-words. Sentences are usually very short.

Disadvantages: Need for specialized vocabularies. Expressions sometimes only understood by special interest groups. Extended disdain for orthography and punctuation rules.

²⁵ Munday, J. (2001), *Introducing Translation Studies: Theories and Applications*; Routledge, Chapter 2.

3.4.2.4 E-Commerce texts

Original text	Our translation
World War I THE GREAT WAR ORIGINAL AUSTRALIAN LIGHT HORSEMAN STUDIO PORTRAIT JAMES P. QUINN - COLLECTIBLE ORIGINAL OIL " PORTRAIT OF AN OLD WOMEN " SIGNED GABRIELE SCHWEITZER-DAUBE GERMAN WC "YOUNG BOY STANDING" 1922 Don Ricks Original Oil Painting American Indian Portrait 1981 Excellent Condit Henry Hanke (1901-89) Original Oil Painting Sunlit Cottage Winter in Windsor Harald Vike (1906-87) Original Oil Painting Afternoon Stroll along Sandgate QLD HIGH WOODEN COVERED TRAIN BRIDGE OVER WINOOSKI RIVER GORGE VERMONT~WM STACY~FINE Elioth Gruner (1882-1939) Original Oil Painting River Landscape with Willows	Primera Guerra Mundial la Gran Guerra retrato original de caballería ligera estudio australiano James P. Quinn- óleo original coleccionable "Retrato de una anciana" firmado Gabriele Schweitzer-Daube WC alemán "Chico joven de pie" 1922 Don Ricks pintura al óleo original retrato indio americano 1981 estado excelente Henry Hanke (1901-89) pintura al óleo original casa de campo bañada por el sol en invierno en Windsor Harald Vike (1906-87) pintura al óleo original paseo por la tarde a lo largo de Sandgate Queensland Puente ferroviario alto de madera sobre el río Winooski Gorge Vermont ~ WM Stacy ~ Bueno Elioth Gruner (1882-1939) pintura al óleo original paisaje río con sauces

Advantages: Texts usually limited in length, because of size of ad descriptions. Need to get to the point and describe concisely what is advertised. In our case, the ads are limited to a length of 80 characters. Very simple text, its style is telegraphic.

Disadvantages: Overabundance of ad-hoc acronyms. Not necessarily conveying syntax rules.

3.4.2.5 Subtitles (film captions)

Original text	Official translation
The Loner You love curry, right? I'll give you more. Hey! Is it good? Hey, ugly. Look at her - Crazy bitch. - I asked you a question! That's better! - Looking good! - That's better! It suits her.	La Solitaria Te encanta el curry, ¿no? Te pondré más. ¡Eh! ¿Está bueno? Eh, fea. Miradla. - Puta loca. - ¡Te he preguntado! ¡Así estás mejor! - ¡Estás muy guapa! - ¡Mucho mejor! Le queda bien.

- Hey.
- What the?

- Eh.
- Mierda.

Advantages: Limited vocabulary. Mimics spoken speech.

Disadvantages: Excess of interjections and short segments, usually with 3 or even less words. This will have a probable influence on BLEU scoring, causing some distortion due to the brevity penalty. Lack of visual context: in a natural conversation, visual context is fundamental and together with language they convey the whole meaning. These translations seem to have been made without screening the film, because sometimes there are gender discrepancies in the adjectives and the text is very plain. This factor is especially flagrant in cases of intermediate translations, with a tendency to lose information (in this case from Korean into English, as intermediate language, and from English into Spanish. Overabundance of repetitions: there are a total of 92 lines duplicated out of the 500 that we have taken to analyse. This could cause some distortion in the final scoring results. The following is the result from the command:

```
$ sort OpenSubtitles2018.en-es.500.en | uniq -cd | sort -nr
```

8 What?	3 Really?	2 Stop it.
7 Soo-na?	3 Please!	2 Stop!
5 Moong-chi!	3 It's me.	2 Soo-na...
4 You bitch!	3 Help me!	2 Open up.
4 Where are you?	3 Get up!	2 Let's eat.
4 Soo-na!	2 You whore!	2 Let it go.
4 Joo-young!	2 You little thief!	2 It's me, Soo-na.
4 Hey!	2 What are you talking about?	2 I have an appointment.
3 Yes.	2 Thanks.	2 H1K1:
3 What are you doing?	2 Stop it, Soo-na!	2 Don't worry.

All these considerations make difficult to figure out whether the translation was made by a professional or simply a fan that made use of translation memories that consistently replicate terms and expressions in the Spanish output as well.

3.4.3 File text formatting

We will arbitrarily select for our purposes the segment of the Treaty of the European Union, between the Fourth Part, XXII Title and the Sixth Part, a total of 453 lines of text. We must emphasize that we have made our own division in sentences, not relying in existing scripts such as **split-sentence.pl**. In the process, we will also assure that the file contains only one sentence (segment) per line. This can be observed in the following first 7 lines of the source text (**treaty-ue.en**); we have employed signs such as '□' and '\$' to visualize non-printable characters in the text, representing the spaces and end of line character, respectively.

```
TITLE□XXII$
TOURISM$
Article□195$
```

1. The Union shall complement the action of the Member States in the tourism sector, in particular by promoting the competitiveness of Union undertakings in that sector.

To that end, Union action shall be aimed at:

(a) encouraging the creation of a favourable environment for the development of undertakings in this sector;

(b) promoting cooperation between the Member States, particularly by the exchange of good practice.

First of all, we will tokenize the text with a command like this:

```
$ tokenizer.perl < treaty-ue.en > tok-treaty-ue.en
```

In our example the words and punctuation will be treated as independent entities, each isolated from the rest by two spaces, one before and one after the identified token.

TITLE XXII

TOURISM

Article 195

1. The Union shall complement the action of the Member States in the tourism sector, in particular by promoting the competitiveness of Union undertakings in that sector.

To that end, Union action shall be aimed at:

(a) encouraging the creation of a favourable environment for the development of undertakings in this sector;

(b) promoting cooperation between the Member States, particularly by the exchange of good practice.

The next step is the conversion of uppercase letters to lowercase.

```
$ lowercase.pl < tok-treaty-ue.en > lc-tok-treaty-ue.es
```

The result of the previous processing will be something like this:

title xxii

tourism

article 195

1. the union shall complement the action of the member states in the tourism sector, in particular by promoting the competitiveness of union undertakings in that sector.

to that end, union action shall be aimed at:

(a) encouraging the creation of a favourable environment for the development of undertakings in this sector;

(b) promoting cooperation between the member states, particularly by the exchange of good practice.

This process is represented in Figure 2 in Annexes. When we have the corpus almost ready (in its English and Spanish versions), we will purge it of void, too short or

too long lines with the script **clean-corpus-n.pl**. In our case this step is possibly²⁶ redundant, but we decided to do it just in case.

```
$ clean-corpus-n.pl lc-tok-treaty-ue en es cleaned 1 100
```

This step will generate two files (**cleaned.en** and **cleaned.es**) ready for score calculation.

Now we will make use of the shell script **procesa-legal.sh** (see section 7.3.1) which will process the files successively to **tokenizer.perl**, **lowercase.perl** and **clean-corpus.pl**. The last step will be the script **score.rb** and a separate command line for Java. The result will be a series of files, each of them related with a specific metric.

File name	Content
legal-googlebleu-out	BLEU of the Google Translate candidate
legal-googleter-out	TER of the Google Translate candidate
legal-googlemeteor-out	METEOR of the Google Translate candidate
legal-opennmtbleu-out	BLEU of the OpenNMT candidate
legal-opennmtter-out	TER of the OpenNMT candidate
legal-opennmtmeteor-out	METEOR of the OpenNMT candidate

For convenience, we put our files in the places where `score.rb` expects them to be. This implied to use the directory named *scoring*, which is generated from the `score.rb` tarball. To keep things apart, we created five subdirectories called *amazon*, *eBay*, *novela*, *subtitulos* and *tratado-ue*. Each of these categories has its own script: **procesa-amazon.sh**, **procesa-ebay.sh** ... and so on.

3.4.4 Obtaining texts to be used as translation references and candidates

Depending on the domain of the text, we have used different methods to obtain the raw texts. Sometimes it simply implies extracting text from PDFs; in other cases, the process is much more involved. As an example, we will show the original format of just one entry between the 231.780 that constitute the small subset employed from Amazon archives.

²⁶ As part of the training of an SMT system, the corpus of aligned (parallel) translated sentences is preprocessed with the `clean-corpus.pl` script. It eliminates (both in the original and in the translation) sentences that are too long or too short, as an input requirement of GIZA++ (Franz Josef Och, Hermann Ney. "A Systematic Comparison of Various Statistical Alignment Models", Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003), a program from the MOSES suite. In our case we are not training an SMT, so it may be absolutely superfluous.

The data is structured, that is, each register follows an internal convention of fields, like a database. In our example, we are visualizing a single register, the structure of which will be made more apparent in the following manner:

```
{
  "reviewerID": "A2HD75EMZR8QLN",
  "asin": "0700099867",
  "reviewerName": "123",
  "helpful": [8, 12],
  "reviewText": "Installing the game was a struggle (because
of games for windows live bugs).Some championship races and
cars can only be \"unlocked\" by buying them as an addon to
the game. I paid nearly 30 dollars when the game was new. I
don't like the idea that I have to keep paying to keep
playing.I noticed no improvement in the physics or graphics
compared to Dirt 2.I tossed it in the garbage and vowed
never to buy another codemasters game. I'm really tired of
arcade style rally/racing games anyway.I'll continue to get
my fix from Richard Burns Rally, and you should to.
:)http://www.amazon.com/Richard-Burns-Rally-
PC/dp/B000C97156/ref=sr_1_1?ie=UTF8&qid;=1341886844&sr;=8-
1&keywords;=richard+burns+rallyThank you for reading my
review! If you enjoyed it, be sure to rate it as helpful.",
  "overall": 1.0,
  "summary": "Pay to unlock content? I don't think so.",
  "unixReviewTime": 1341792000,
  "reviewTime": "07 9, 2012"
}
```

Of all the above fields, we will be only interested in the one marked as “reviewText”. This kind of text will require a lot of manual pre-processing, because of the sometimes-confusing sentence boundaries and the prolific use of apostrophes. We used the Python **lee-comentarios.py** program, which extracts from each register the reviewText field and writes it on standard output. The problem which such simple program is that it generates a single line 47.661.132 words long, were all the reviewText fields are simply concatenated one after the other, which is unusable without further processing. As we use for our test sets only about 400-500 lines, we split this single enormous line into its constituent’s original lines with the use of a very powerful text editor: Joe²⁷. It has a simple alphanumeric interface but its real strength is the capability to work on text files no matter their size, which was crucial in our case.

The search of on-line eBay entries was easier (**busca-multiples-claves.py**), as the algorithm produced directly the text entries, requiring no further processing.

²⁷ <https://joe-editor.sourceforge.io/>. JOE is being maintained by its original author Joseph Allen, plus all of the people who send bug reports, feature suggestions and patches to the project web site. JOE is hosted by SourceForge.net and its source code is controlled under Mercurial.

4 RESULTS

We will submit the pre-processed text belonging to the four domains to the **score.rb** script, which works perfectly except as noted with the METEOR score. Having independently obtained this last metric, the results are the following:

Domain of text	MT employed	Acronym	score.rb			METEOR 1.4
			BLEU	NIST	TER	
Legal	Google Translate	LG	0.5456	8.6245	0.3202	0.4118
	OpenNMT	LO	0.5527	8.7374	0.3110	0.4138
Amazon	Google Translate	AG	0.5335	8.4233	0.3185	0.4040
	OpenNMT	AO	0.4749	7.8255	0.3679	0.3765
eBay	Google Translate	EG	0.4879	7.3215	0.3661	0.3857
	OpenNMT	EO	0.3551	5.9942	0.5431	0.2901
Novels	Google Translate	NG	0.1529	4.8051	0.7283	0.2251
	OpenNMT	NO	0.1313	4.4147	0.7622	0.2089
Subtitles	Google Translate	SG	0.2684	4.8008	0.5254	0.3165
	OpenNMT	SO	0.1895	3.642	0.7573	0.2762

In these examples, BLEU is consistently better than METEOR, except when it comes to evaluating literature.

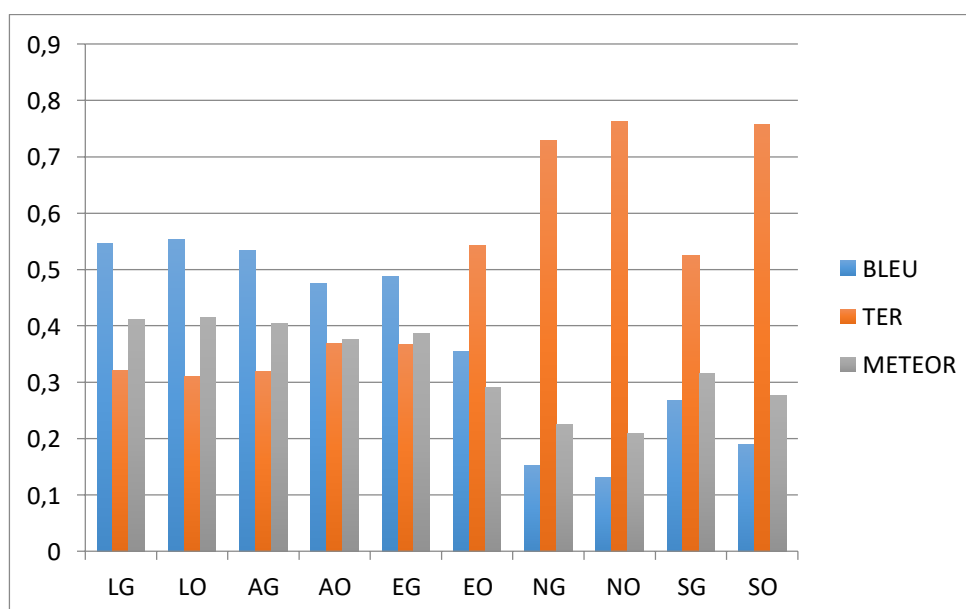


Illustration #5. BLEU, TER and METEOR

If we compare between Google Translate and OpenNMT, the conclusions will be much domain-dependent. BLEU is consistently higher for Google Translate, except significantly for the legal domain, where the two are almost identical. This may reflect that both systems must have fed on exactly the same legal corpora and so their behaviour is similar. In the other explored domains, Google Translate surpasses clearly OpenNMT.

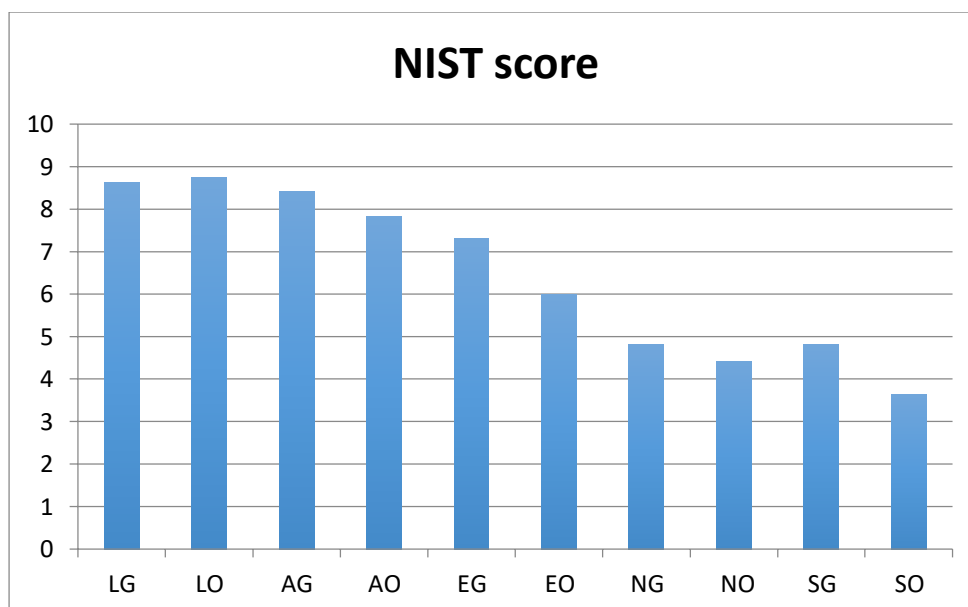


Illustration #6. NIST

As would be expected, BLEU and TER exhibit an inverse relationship. For high BLEU scores, the number of translator edits (TER) will be lower, and the inverse will remain true. In the literary domain, where conventional scores are down, TER will arise as the most significant metric. The same behaviour described above for BLEU is observed with the related NIST metric.

As we already know, BLEU and METEOR are the higher the better and each of them attends to different aspects; on the contrary, TER has the opposite meaning: the lowest, the better. If we combine this three different metrics into one single variable, it could be argued that the best translation would be the one where the sum of BLEU, TER and METEOR is highest. In the following bar chart, LG, LO and AG are on a par as the best overall translation quality measured.

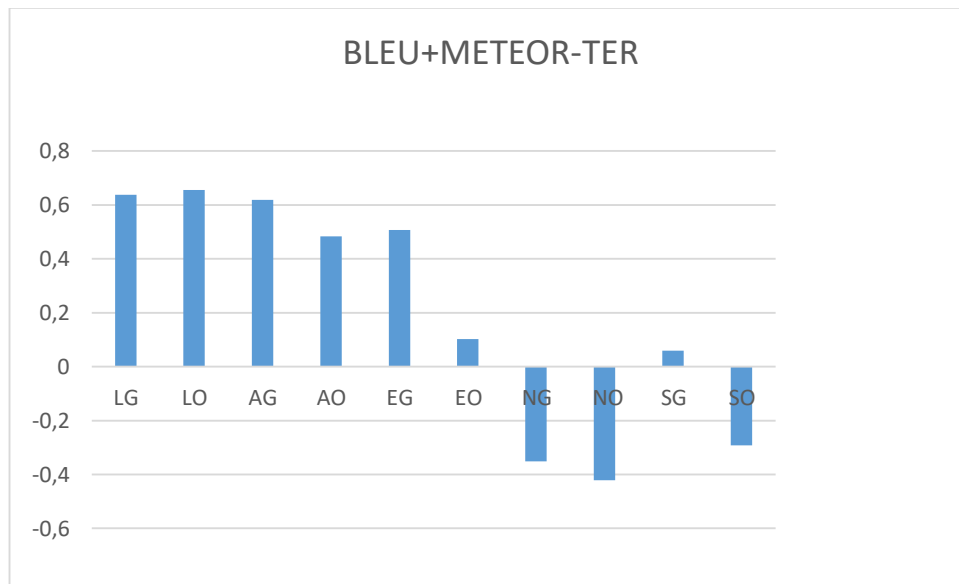


Illustration #7. BLEU+METEOR-TER

As we have shown, using a script (**score.rb**) that almost handles everything has great advantages, but even if it is relatively recent (last updated in 2010), the MT field evolves at great speed and is in the danger of getting outdated. In a recent publication (Post M., 2018), its author addresses problems arising from the fact that user-supplied reference tokenization is a source of incompatibility. Pre-processing involves several steps, such as normalization, tokenization, compound-splitting, removal of case, etc. The author identifies the tokenization of the references as the most critical of them; if the references actually used between experiments differ, the results are compromised.

In order to avoid this, the author provides a new tool called SacreBLEU²⁸, which computes two scores, BLEU and CHRF, being the former the default. For our purposes, we will feed our user-processed translation candidates to SacreBLEU, providing the reference translation unprocessed. In the computation of the CHRF score, we leave the default n-gram length since it equals 6, an optimum value (Popvic, M., 2015).

As an example, use of SacreBleu on texts obtained from Amazon would be like this:

```
$ cat C1-amazongoogle.es | sacrebleu amazon.es > resultados/amazon-google.bleu
```

```
$ cat C1-amazongoogle.es | sacrebleu --metrics=chrf --chrf-beta=3 amazon.es > resultados/amazon-google.chrf3
```

²⁸ <https://github.com/awslabs/socketeye/tree/master/contrib/sacrebleu>

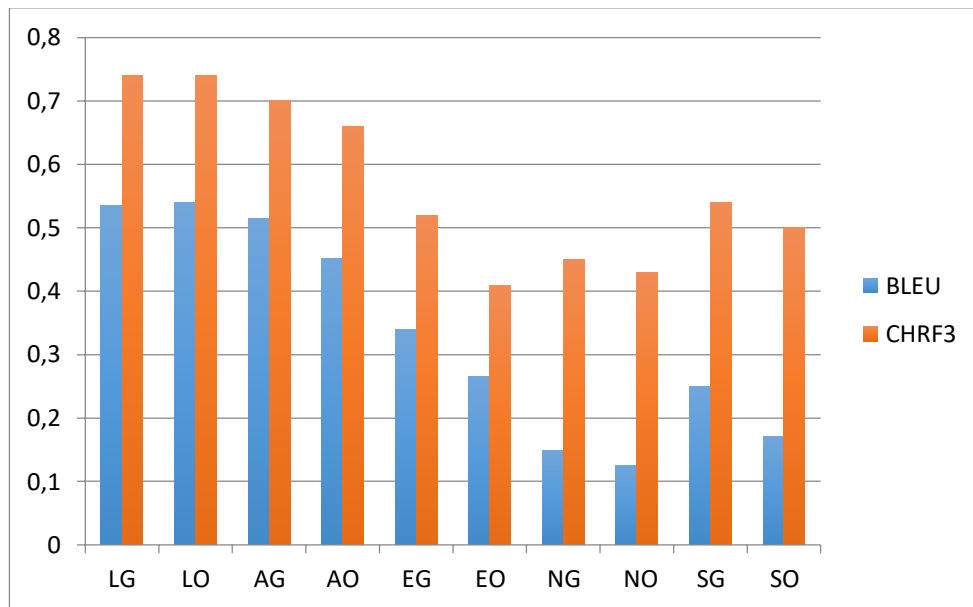


Illustration #8. BLEU vs CHR3

When we compare the BLEU scores computed with the two scripts, we find that **score.rb** values are always higher, being the differences usually subtle, except in the case of the eBay listings. In this case they could be attributed to subtle variations in the manual pre-processing of the candidate files, but we must also take into consideration that the two scripts employ variations of the same program for computing the BLEU score:

- mteval-v13m.pl score.rb
- mteval-v13a.pl sacreBLEU

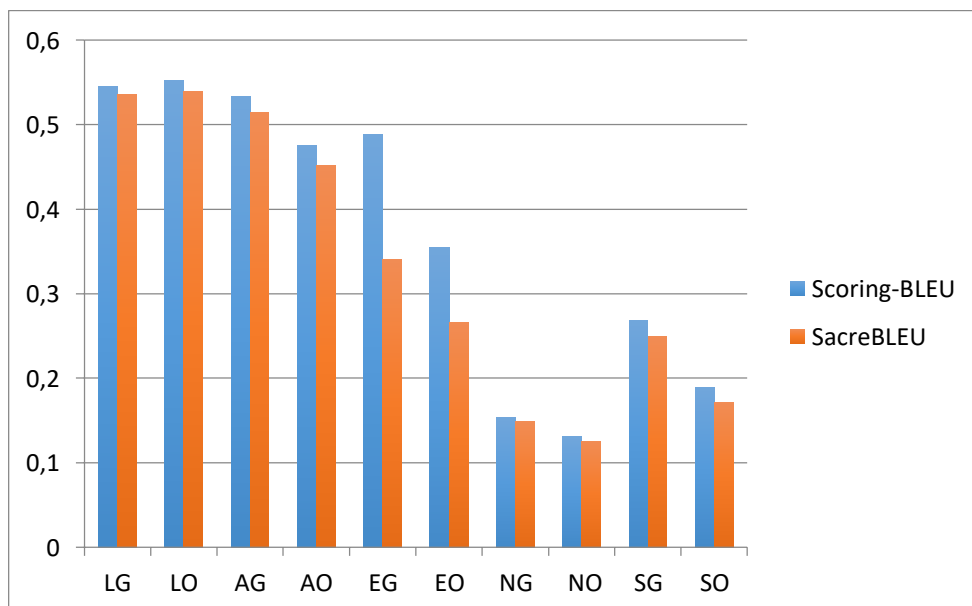


Illustration #9. Conventional BLEU vs SacreBLEU

It could be deduced that the computational differences between the two²⁹ are exacerbated with the peculiarities of eBay titles.

The above data compares the behaviour of the different corpora with the estimators considered. It is also worthy of note the magnitude of the difference between the output of Google Translate and OpenNMT. If the value is positive, GT fares better than ONMT; if it's negative, the opposite is true. Except with the Amazon comments, the biggest difference stands with the TER estimate.

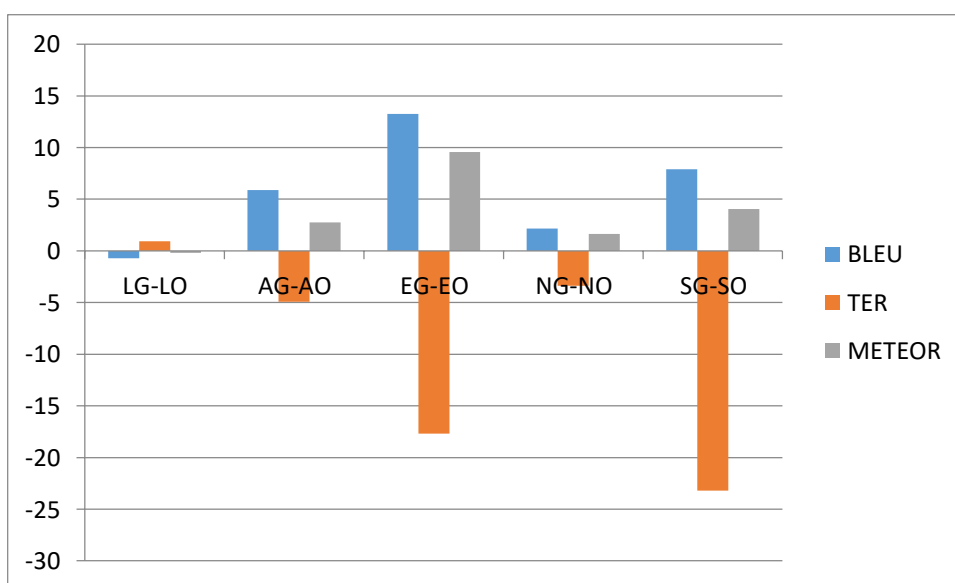


Illustration #10. Magnitude of the discrepancy between Google Translate and OpenNMT with the different corpora analysed

From the above bar chart, we can observe that in the same linguistic category the lesser the dispersion is, the better is the estimator for that category. On the one hand, in the case of legal texts, it's evident that the BLEU, TER and METEOR estimates differ only marginally between the Google Translate and OpenNMT translations. On the other hand, in the eBay and Subtitles categories, the extreme variation points to significant deviations of the translations obtained from Google Translate and OpenNMT from the text references employed.

²⁹ When running the different test sets with score.rb, we always get the same, non-fatal message, possible indication that the code of mteval-v13m.pl is not up-to-date with the current syntax of Perl. The text of the warning message is the following: Use of 'Hyphen' in \p{} or \P{} is deprecated because: Supplanted by Line_Break property values; see www.unicode.org/reports/tr14; at /home/usuario/Dropbox/traduccion/scoring/mteval-v13m.pl line 960.

5 CONCLUSIONS

The stated purpose of this study is the one expressed in its title: “Comparison of the adequacy of on-line translations with different genres of texts”. The measure of the adequacy has been made employing well-known quality estimators: BLEU, TER, METEOR... The estimators have been employed over small samples (aprox. 500 lines each) of the types of texts chosen, and in a single run, so we don't provide an estimation of the statistical validity of the results. Concerning the types of texts, we have settled for five categories: legal, comments in web pages (Amazon), e-commerce (ads in eBay), literature and film subtitles.

There is an enormous variety of texts available on the internet and readily-available on-line translation machines that can be used for the translation of these or other texts; of those, we have employed Google Translate (GT) and Openmt (OP).

The different types of texts have been compared with the translations these on-line tools offer, and the conclusions are not clear-cut. Everyone can reach his own conclusions, but we will try to offer our own, with the limitations mentioned before.

Optimal translations (from the point of view of the estimators considered) could be those where high scores of estimators like BLEU and METEOR coexist with low scores on estimators such as TER (shown in illustration #5). That was the case with the legal texts, where GT and OP went on a par, with a slight advantage of OP. Next came the translations of Amazon Comments, where GT performed noticeably better. In the rest of categories, the quality of translation was severely degraded (as pointed out by the high values of TER), due to several factors intrinsic to the nature of the translated texts.

How much does the quality of translation differ for GT and OP given a certain category of text? This can be visualized in illustration #10, where a narrow range of variation between the score bars expresses little variation in the quality in the translations of a given text (case of legal but also surprisingly of literature). On the opposite side, eBay ads and Subtitles showed the greatest disparity between GT and OP translations.

6 REFERENCES

“findItemsAdvanced”. eBay developers program. eBay Finding API Version 1.13.0 (<https://developer.eBay.com/devzone/finding/callref/finditemsadvanced.html#Request.descriptionSearch>) [visited on 04/14/2018]

Babych, B., “Weighted N-gram model for evaluating Machine Translation output”, Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics, 2004, 15-22

Brownlee, J., “Deep Learning for Natural Language Processing. Develop Deep Learning Models for Natural Language in Python”, (2018) (E-book, edition v1.2). Especially chapter 24.

Castilho, S., Moorkens, J., Gaspari, F., Iacer Calixto, I., Tinsley, J., Way, A., “Is Neural Machine Translation the New State of the Art?”, The Prague Bulletin of Mathematical Linguistics, number 108 June 2017, 109-120.

Denkowski, M., Lavie, A., “Meteor universal: Language specific translation evaluation for any target language”, Proceedings of the 9th Workshop on Statistical Machine Translation (WMT), 2014

Doddington, G., “Automatic evaluation of machine translation quality using n-gram co-occurrence statistics”, Proceedings of the second international conference on Human Language Technology Research, San Diego 2002, 128-132.

Heafield, K., Lavie, A., “Combining Machine Translation Output with Open Source. The Carnegie Mellon Multi-Engine Machine Translation Scheme”, The Prague Bulletin of Mathematical Linguistics, 93 January 2010, pp. 27–36

Hwang, M., "Computing BLEU scores" (<http://ssli.ee.washington.edu/~mhwang/pub/loan/bleu.pdf>) [visited on 04/14/2018]

Lavie, A., Sagae, K., Jayaraman, S., “The Significance of Recall in Automatic Metrics for MT Evaluation”, Proceedings of the 6th Conference of the Association for Machine Translation in the Americas (AMTA-2004).

McAuley, J., “Amazon product data” (<http://jmcauley.ucsd.edu/data/amazon/>) [visited on 04/14/2018]

Papineni, K., Roukos, S., Ward, T., Zhu W., “BLEU: a Method for Automatic Evaluation of Machine Translation”. IBM Computer Science, September 17 (2001)

Popvic, M. "CHRF: character n-gram F-score for automatic MT evaluation", Proceedings of the Tenth Workshop on Statistical Machine Translation, pp. 392-395. Lisboa 2015

Popvic, M., "CHRF deconstructed: β parameters and n-gram weights", Proceedings of the First Conference on Machine Translation, vol. 2, pp. 499-504, Berlin 2016.

Post, M. "A Call for Clarity in Reporting BLEU Scores" (arXiv:1804.08771 [cs.CL])

Slocum, J., "A Survey of Machine Translation: Its History, Current Status and Future Prospects", Computational Linguistics, Volume 11, Number 1, January-March 1985

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J., "A study of translation edit rate with targeted human annotation", Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Visions for the Future of Machine Translation (AMTA-06), Cambridge 2006, 223–231.

Wołk K., Korzinek D., "Comparison and Adaptation of Automatic Evaluation Metrics for Quality Assessment of Re-Speaking", PHD Workshop WDSIT2015, 18-20 October 2015, Kazimierz Dolny, Polska

Wołk K., Marasek K., "Enhanced Bilingual Evaluation Understudy", Lecture Notes on Information Theory, ISSN: 2301-3788, 2014

7 ANNEXES

7.1 Figures

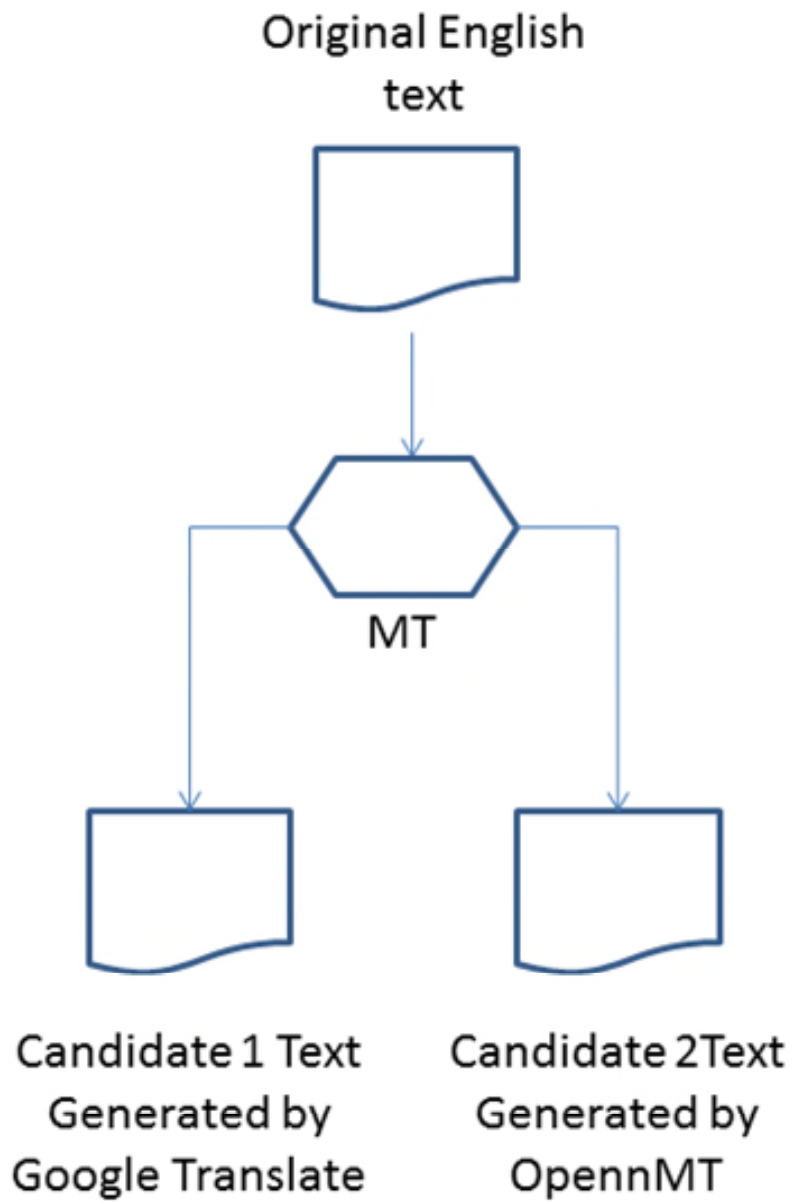


Figure 1. Obtaining candidate translations of one source text

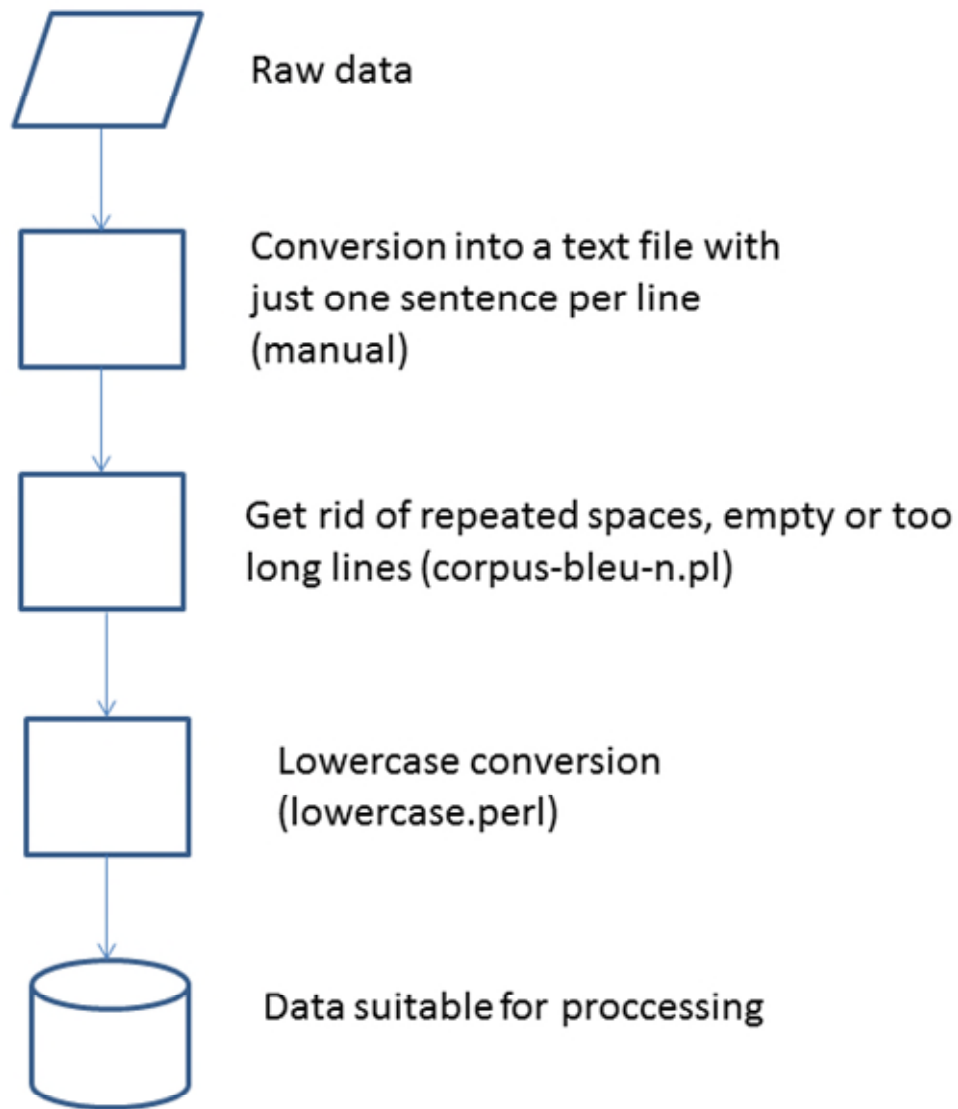


Figure 2. MOSES-style formatting of text previous to calculations

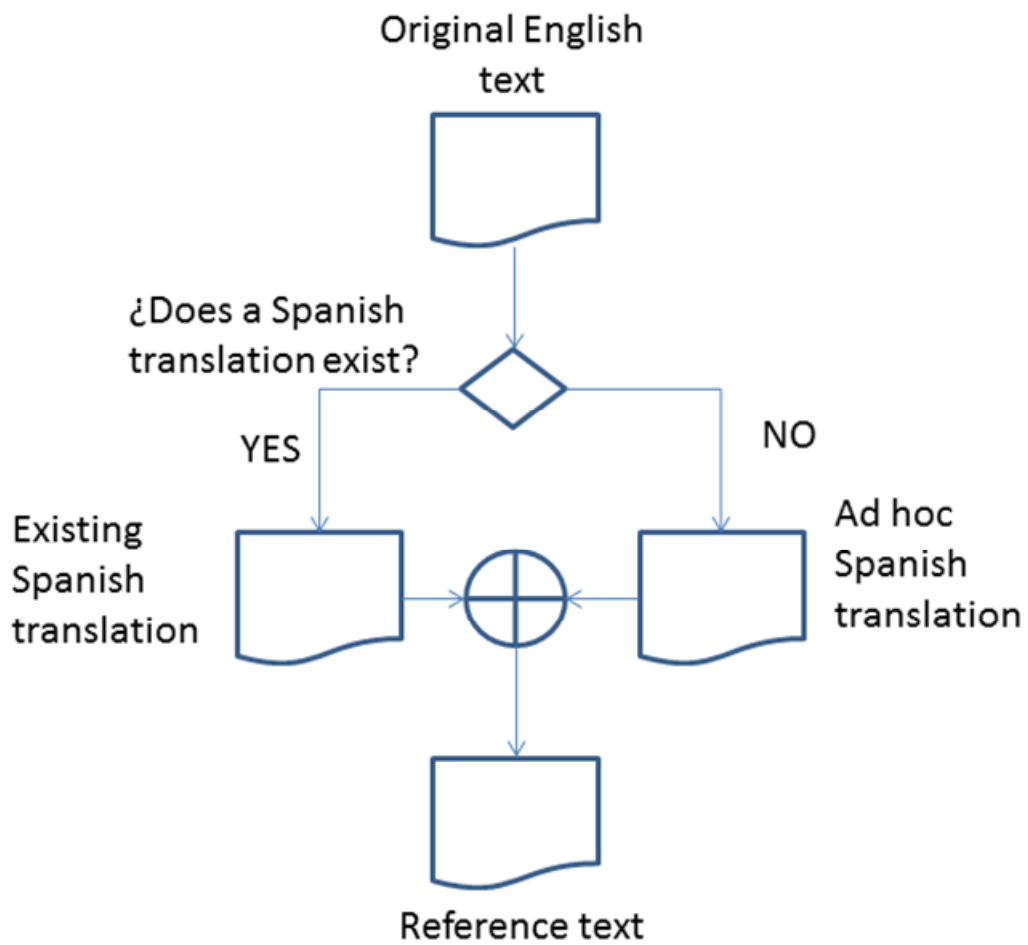


Figure 3. How to obtain translations whose purpose is to serve as references.

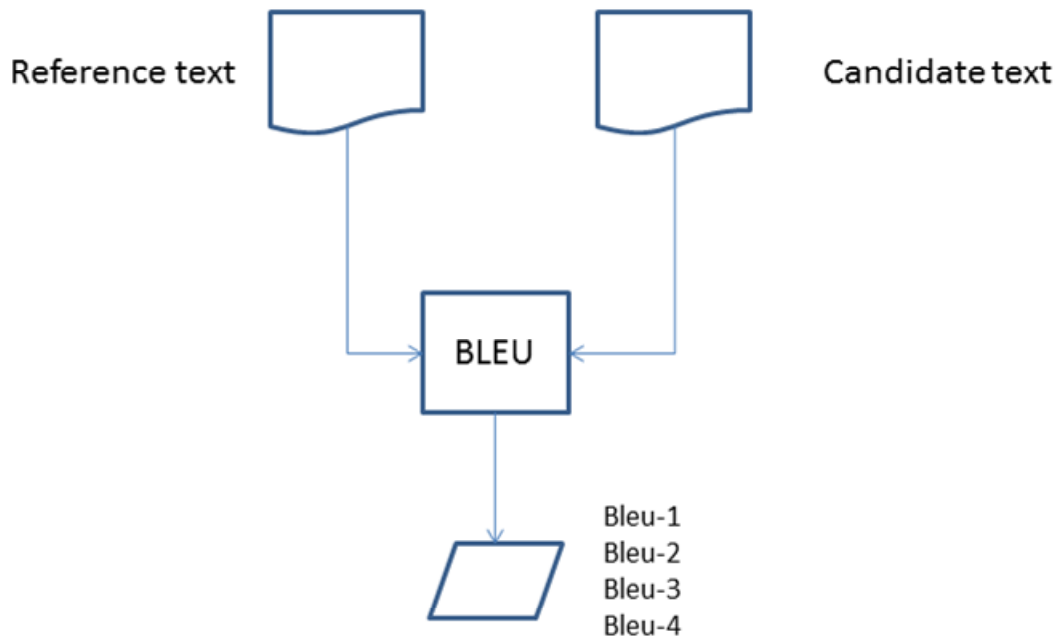


Figure 4. Inputs provided to the BLEU algorithm

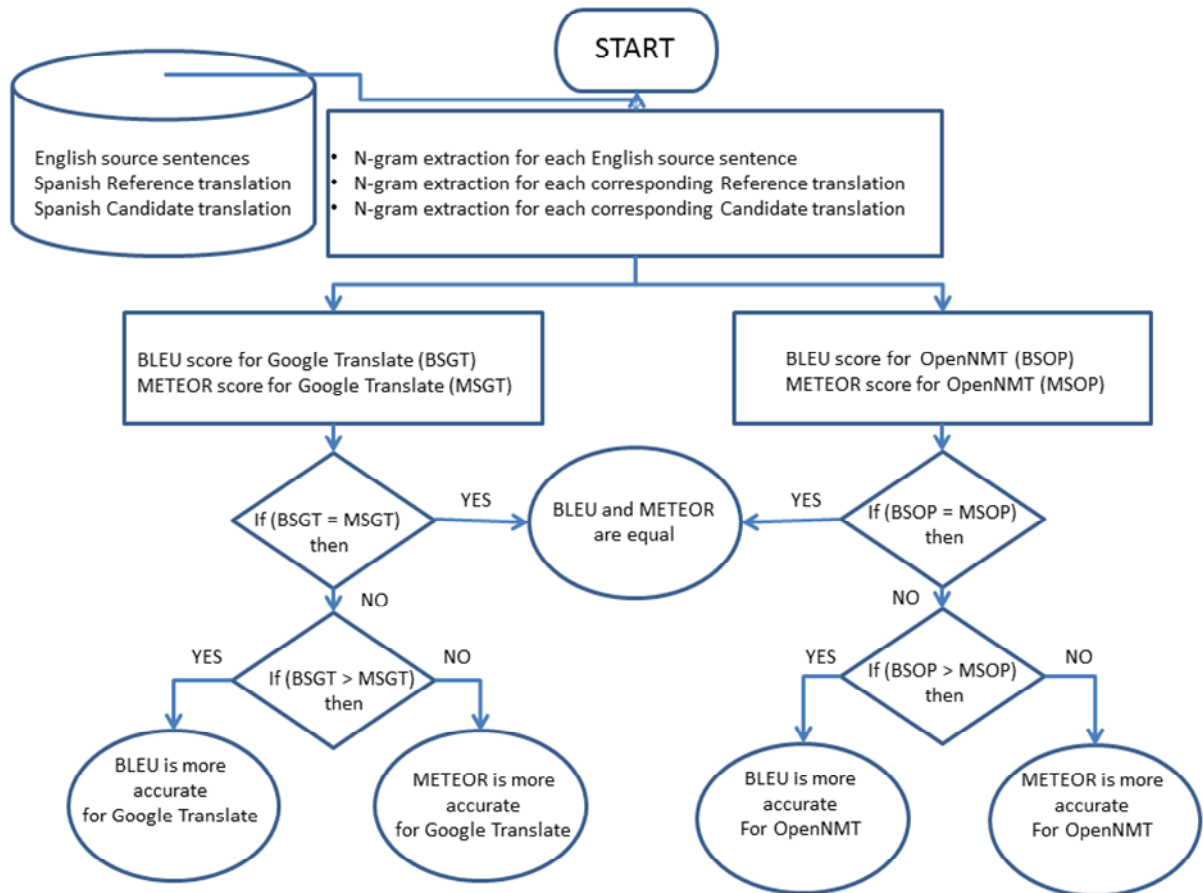


Figure 5. Comparison between two translation adequacy metrics (BLEU and METEOR). Inspired in figure 1 of (Hadla, L.S. et al, 2015)

7.2 Code

7.2.1 procesa-legal.sh

```
#!/bin/bash
# Script realizado en shell que realiza el preprocesamiento de los textos legales.
# Pretende recoger detalladamente todos los pasos, para facilitar
# trabajar con tantos ficheros.
# Está pensado para ser ejecutado desde el directorio "scoring"

# TEXTOS LEGALES

echo "Preparando Textos Legales: candidato Google Translate..."
# Procesado de candidato Google Translate y del fichero ingles del cual deriva
  programas/tokenizer.perl < tratado-ue/es-google/C1-google.en > tratado-ue/es-
google/tok-C1-google.en
  programas/tokenizer.perl < tratado-ue/es-google/C1-google.es > tratado-ue/es-
google/tok-C1-google.es

  programas/lowercase.perl < tratado-ue/es-google/tok-C1-google.en > tratado-ue/es-
google/lc-tok-C1-google.en
  programas/lowercase.perl < tratado-ue/es-google/tok-C1-google.es > tratado-ue/es-
google/lc-tok-C1-google.es

  programas/clean-corpus-n.pl tratado-ue/es-google/lc-tok-C1-google en es tratado-ue/es-
google/cleaned 1 100

# Limpiamos los ficheros intermedios
  rm tratado-ue/es-google/tok-* tratado-ue/es-google/lc-tok-*

echo
echo "Preparando Textos Legales: candidato Opennmt..."
# Procesado de candidato OpenNMT y del fichero ingles del cual deriva
  programas/tokenizer.perl < tratado-ue/es-opennmt/C2-opennmt.en > tratado-ue/es-
opennmt/tok-C2-opennmt.en
  programas/tokenizer.perl < tratado-ue/es-opennmt/C2-opennmt.es > tratado-ue/es-
opennmt/tok-C2-opennmt.es

  programas/lowercase.perl < tratado-ue/es-opennmt/tok-C2-opennmt.en > tratado-ue/es-
opennmt/lc-tok-C2-opennmt.en
  programas/lowercase.perl < tratado-ue/es-opennmt/tok-C2-opennmt.es > tratado-ue/es-
opennmt/lc-tok-C2-opennmt.es

  programas/clean-corpus-n.pl tratado-ue/es-opennmt/lc-tok-C2-opennmt en es tratado-
ue/es-opennmt/cleaned 1 100

# Limpiamos los ficheros intermedios
  rm tratado-ue/es-opennmt/tok-* tratado-ue/es-opennmt/lc-tok-*

# CALCULO DE SCORES
# BLEU, NIST, TER (METEOR no funciona y se invoca separadamente)
echo "#####"
echo "Procesando Textos Legales traducidos por Google Translate: cálculo de scores BLEU,
NIST y TER"
./score.rb --language es --hyp tratado-ue/es-google/C1-google.es --ref tratado-ue/es/treaty-
```



```

for l in g:
    yield eval(l)

comentarios = []

for review in parse("reviews_Video_Games_5.json.gz"):
    comentarios.append(review['reviewText'])

print (comentarios)

```

7.2.3 busca-multiples-claves.py

```

# Programa Python que busca cuadros con una amplia variedad
# de temáticas en la web de eBay UK. Requiere registrarse como
# programador de eBay para poder tener acceso.

from eBaySdk.finding import Connection

api = Connection(siteid='EBAY-GB', appid='XXXX-Encuentr-PRD-77879bc50-a9bf32ec',
config_file=None)

# la descripción de esta función y sus opciones la tomo de:
#https://developer.eBay.com/devzone/finding/callref/finditemsadvanced.html#Request.descriptionSearch

# Lista de keywords que quiero listar. Están escogidas por tanteo y error; si lo que
# suponemos pueda ser una keyword devuelve algo de la base de datos,
# la incorporamos a la lista
# Keywords probadas:
# Nude, Portrait, Landscape, Children, Dog, Women, Girl, Death, Moon, Still
# Jesus, Christ, King, Cross, War, Church, Ship, Town, Castle, Soldier,Hunting
claves = ['Nude', 'Portrait', 'Landscape', 'Children', 'Dog', 'Women', 'Girl', 'Death',
'Moon','Still','Jesus', 'Christ', 'King', 'Cross', 'War', 'Church', 'Ship', 'Town', 'Castle',
'Soldier','Hunting']

for clave in claves:
    api.execute('findItemsAdvanced', {
        'keywords': clave,
        'descriptionSearch': 'true', # If true, the text of the item's description and subtitles
# will be included in the search. If false, only item titles are included in keyword searches.
        'categoryId': ['551'], # Specifies the category from which you want to retrieve item listings.
# This field can be repeated to include multiple categories.
# Up to three (3) categories can be specified.
        'itemFilter': [ # Reduce the number of items returned by a find request using item filters.
# Use itemFilter to specify name/value pairs.
# You can include multiple item filters in a single request.
            {'name': 'Condition', 'value': 'Used'},
            {'name': 'MinPrice', 'value': '200', 'paramName': 'Currency', 'paramValue': 'GBP'},
            {'name': 'MaxPrice', 'value': '4000', 'paramName': 'Currency', 'paramValue': 'GBP'}
        ],
        'paginationInput': {
            'entriesPerPage': '25',
            'pageNumber': '1'
        },
        'sortOrder': 'CurrentPriceHighest'
    })

```

```

})
dictstr = api.response.dict()
for item in dictstr['searchResult']['item']:
    print('{}'.format(item['title']))

```

7.2.4 bleu-ss.py

```

#!/usr/bin/env python
# Programa Python que lee los distintos ficheros de entrada
# (referencia, candidato) y genera en su salida estandar el
# cómputo de sentence_bleu para cada una de las parejas
# referencia-candidato. Adapta el ejemplo descrito en el
# e-book "Machine Learning Mastery with Python" de Jason
# Brownlee de trabajo con cadenas de caracteres a un modelo que
# procese ficheros completos.
# Realiza los procesamientos previos necesarios del texto
# leído de los ficheros antes de pasarlo como argumento
# a la funcion sentence_bleu. No obstante, y para mantener
# cierta compatibilidad, se recomienda procesar antes los
# ficheros con el programa clean-corpus-m.pl de MOSES.
# La salida del programa se regula mediante la variable 'verboso'.

from nltk.translate.bleu_score import sentence_bleu
import sys, getopt

def main(argv):
    referencefile = " # traducciones de referencia
    candidatefile = " # traducciones candidatas
    excludefile = " # (opcional) palabras a excluir

    # el siguiente parámetro puede ser 0 o 1
    # 0 es salida tersa
    # 1 es salida abundante
    verboso = 0

    # procesado de argumentos en linea de comandos
    try:
        opts, args = getopt.getopt(argv,"hr:c:",["rfile=", "cfile="])
    except getopt.GetoptError:
        print (' bleu-s.py -r <reference file> -c <candidate file>')
        sys.exit(2)
    for opt, arg in opts:
        if opt == '-h':
            print (' bleu-s.py -r <reference file> -c <candidate file>')
            sys.exit()
        elif opt in ("-r", "--rfile"):
            referencefile = arg
        elif opt in ("-c", "--cfile"):
            candidatefile = arg

    # intentamos abrir los ficheros proporcionados como argumentos
    # y si no existen nos quejamos y abortamos ejecucion del programa
    try:
        candidate_file = open(candidatefile,'r')
    except OSError as err:
        print("OS error: {0}".format(err))

```

```

    sys.exit(1)

try:
    reference_file = open(referencefile,'r')
except OSError as err:
    print("OS error: {0}".format(err))
    sys.exit(1)

# comienza procesamiento de ficheros de entrada
# leo sus renglones de uno en uno
candidate = candidate_file.readline()
reference = reference_file.readline()
# paso lo leído a minúsculas
candidate = candidate.lower()
reference = reference.lower()
# elimino todos los signos de puntuación
candidate = re.sub('[¿?¡!.,;:\-\(\)\'«»]', '', candidate)
reference = re.sub('[¿?¡!.,;:\-\(\)\'«»]', '', reference)
# elimino el carácter '\n' de final de línea
candidate = candidate.strip('\n')
reference = reference.strip('\n')
if not verboso:
    print("{};{};{};{};{}".format("Referencia","Candidato","B1","B2","B3","B4"))

# mientras no se terminen los renglones en el fichero de traducción
# candidata
while candidate:
# descompongo los renglones en palabras
    words_in_candidate = candidate.split(' ')
    words_in_reference = reference.split(' ')

# cálculo del estimador bleu
    score_bleu_1 = sentence_bleu(reference, candidate, weights=(1, 0, 0, 0))
    score_bleu_2 = sentence_bleu(reference, candidate, weights=(0.5, 0.5, 0, 0))
    score_bleu_3 = sentence_bleu(reference, candidate, weights=(0.33, 0.33, 0, 0))
    score_bleu_4 = sentence_bleu(reference, candidate, weights=(0.25, 0.25, 0.25, 0.25))
    if verboso:
        print("REFERENCE: ", words_in_reference)
        print("CANDIDATE: ", words_in_candidate)
# la salida que recoge el resultado del cálculo consiste
# en una cadena del tipo
# R X, C Y, PUNTUACION: Z
# donde X es la longitud de la lista de referencia,
# Y es la longitud de la lista candidata
# Z es la puntuación bleu obtenida
    print("R ",len(words_in_reference),",", "C ",len(words_in_candidate),",", "B1: ",
score_bleu_1, ", ", "B2: ", score_bleu_2, ", ", "B3: ", score_bleu_3, ", ", "B4: ", score_bleu_4)
    else: # si queremos salida para fichero CSV
        # (Comma Separated Values)
        score_bleu_1 = sentence_bleu(reference, candidate, weights=(1, 0, 0, 0))
        score_bleu_2 = sentence_bleu(reference, candidate, weights=(0.5, 0.5, 0, 0))
        score_bleu_3 = sentence_bleu(reference, candidate, weights=(0.33, 0.33, 0, 0))
        score_bleu_4 = sentence_bleu(reference, candidate, weights=(0.25, 0.25, 0.25, 0.25))

print("{};{};{};{};{}".format(len(words_in_reference),len(words_in_candidate),score_bleu_1,score_bleu_2,score_bleu_3, score_bleu_4))

# repito lectura de renglón
    candidate = candidate_file.readline()

```



```
reference = reference_file.readline()
# paso lo leído a minúsculas
candidate = candidate.lower()
reference = reference.lower()
# elimino todos los signos de puntuación
candidate = re.sub('[¿?¡!.,;:\-\(\)\'«»]', '', candidate)
reference = re.sub('[¿?¡!.,;:\-\(\)\'«»]', '', reference)
# elimino el caracter '\n' de final de línea
candidate = candidate.strip('\n')
reference = reference.strip('\n')
#cierro archivos abiertos
candidate_file.close()
reference_file.close()

if __name__ == "__main__":
    main(sys.argv[1:])
```

7.3 Zusammenfassung

Ziel dieser Diplomarbeit ist es, mithilfe von bereits vorhandenen Schätzfunktionen zur Ermittlung der Angemessenheit die Qualität von maschinellen Übersetzungen³⁰ in Bezug auf Texte aus verschiedenen Themenbereichen zu bewerten. Wir haben beschlossen, die Bereiche Recht, Handel, Spiele, Literatur und Untertitel einzubeziehen, da jeder davon unterschiedliche Merkmale aufweist, sowie verschiedene Schätzfunktionen (BLEU, NIST, METEOR, TER y CHRF) anzuwenden. Jede Schätzfunktion verfolgt einen anderen Ansatz, mithilfe dessen die Qualität der Übersetzungen in jedem einzelnen dieser sprachwissenschaftlicher Bereiche ermittelt werden kann. Das heißt, es soll quantifiziert werden, wie zutreffend die Übersetzungen sind, die mithilfe der gewählten Übersetzungsmotoren, in diesem Fall GoogleTranslate und PureNeural Machine Translation (OpenNMT), durchgeführt wurden.

Hierzu ist zu bemerken, dass die Programmierarbeiten an normalen PCs, ohne besondere Ressourcen, mit SO Linus mit Skripts von Dritten in den Programmiersprachen Ruby, Python y Perl. vorgenommen wurden. Um sie zu manipulieren, griffen wir auf einfache BASH-Skripts zurück(Benutzeroberfläche von Linux).

Der erste Schritt bestand in der Auswahl der zu übersetzenden Texte. In dieser Phase traten bereits Probleme auf: Da es sich um Texte vielfältiger Quellen handelte, lag jeder Text in einem anderen Format oder einem anderen Dokumententyp vor. In einigen Fällen musste der PDF-Text nur manuell extrahiert und in ein Textformat importiert werden. Für die Beispieltex te aus dem Bereich Literatur war es erforderlich, Fragmente zu suchen, die aus irgendeinem Grund in der Public Domain verfügbar waren. In anderen Fällen, wie bei Anzeigen von eBay oder Kommentaren von Amazon, musste auf bereits bestehende Algorithmen-Bibliotheken zugegriffen werden, um nur die Fragmente von Interesse zu extrahieren. Es galt, den unterstützenden Webcode zu ignorieren.

Im zweiten Schritt galt es, die Referenzübersetzungen anzufertigen. Dafür wurden Originaltexte auf Englisch ausgewählt, die nach Möglichkeit eine offizielle Version auf Spanisch mitlieferten. Die Texte, für die keine offizielle Version auf Spanisch verfügbar war, wurden von menschlichen Übersetzern bearbeitet und dann als Referenzübersetzungen verwendet. Die Übersetzungen wurden mithilfe des Microsoft Übersetzers Bing angefertigt, um den Prozess zu beschleunigen. Die notwendigen Korrekturen wurden dann von einem Linguisten vorgenommen.

Im Anschluss verglichen wir die Referenzübersetzungen mit den Zielübersetzungen, die durch die Eingabe des englischen Originaltextes in die gewählten automatisierten

Übersetzungstools, Google Translate³¹ und PureNeural Machine Translation³² (OpenNMT) (c) angefertigt wurden.

Nachdem die Originaltexte, die Referenzübersetzungen und die Zielübersetzungen vorlagen, mussten die Texte anhand der Auswertung von BLEU, TER und anderen Parametern beurteilt werden. Hierbei sollte erwähnt werden, dass wir in dieser Phase die BLEU-Berechnung zuerst manuell mithilfe einfacher mathematischer Formeln durchführten, um den Prozess zu verstehen. Dann wurde versucht, ein Programm zu erstellen, das durch Zugriff auf die NLTK-Bibliothek nur den BLEU-Parameter berechnen sollte. Angesichts der Schwierigkeiten, die sowohl mit der Programmierung als auch mit der Vergleichbarkeit von mit Standardtools erstellten Ergebnissen einhergingen, setzten wir auf Programme, die die Parameter zusammen berechnen würden. So konnten wir vermeiden, jedes Mal für die Errechnung jedes einzelnen Indikators ein anderes Programm suchen und verwenden zu müssen. Somit konnten wir viel Zeit und Arbeit sparen. Das für diese Errechnung zuständige Programm `score.rb`, ist ein leistungsfähiges Instrument aus dem Internet (github). Zur Durchführung der Errechnung müssen diesem Programm nur die Dokumente so zur Verfügung gestellt werden, damit es sie bearbeiten kann: Die Texte haben eine Reihe von gemeinsamen Merkmalen zu erfüllen und müssen in einer übersichtlichen Ordnerstruktur organisiert sein. In diesen Ordnern müssen sich Quelltexte, Referenzübersetzungen und zu bewertende Parallel-Zielübersetzungen befinden. Die zu verwendenden Texten müssen in Dateien mit einem Satz pro Zeile gespeichert werden, die mit dem Zeichen für neue Zeile (`\n`) enden. Zwischen den verschiedenen Dateien muss Zeile für Zeile eine Entsprechung bestehen.

Nach Erfüllung dieser wichtigen Voraussetzungen müssen an den Texten eine Reihe von Veränderungen vorgenommen werden. Dieser Prozess wird Vorbereitung genannt. Aufgrund des geringen Umfangs der verwendeten Textbeispiele (Zwischen 300 und 500 Zeilen) nahmen wir die wichtigste Bearbeitung manuell vor. Danach verwendeten wir weit verbreitete Standardprogramme der MÜ, um die Kompatibilität der Ergebnisse sicherzustellen³³. Auf diese Weise wurde Textdateien mit UTF-8 Zeichenkodierung mit derselben Zeilenanzahl erstellt, wobei jede Zeile der Übersetzung der jeweiligen Zeile aus dem englischen Originaltext gegenübergestellt wurde. Dabei wurden Satzzeichen und Kleinschreibung außen vor gelassen.

Nach einer solchen Bearbeitung der Texte kann das Programm `score.rb` zum Einsatz kommen und numerische Ergebnisse der beliebtesten Schätzfunktionen liefern. Diese Ergebnisse werden anschließend in Excel übertragen, um in Grafiken oder Balkendiagrammen aufbereitet zu werden. Mithilfe visueller Darstellungen lassen sich schneller Schlussfolgerungen ziehen.

³¹ <https://translate.google.es/?hl=es>

³² <https://demo-pnmt.systran.net/production#/translation>

³³ `bleu-ss.py`

In Anbetracht der Ergebnisse lässt sich leicht feststellen, dass jeder Text jeweils eine spezifische wieder erkennbare Spur in den Schätzfunktionen hinterlässt, die im größeren oder kleineren Maße den Differenzierungsmerkmalen der verwendeten Texte zugeordnet werden können.