Does AI Have a Sense of Humor? CLEF 2023 JOKER Tasks 1, 2 and 3: Using BLOOM, GPT, SimpleT5, and More for Pun Detection, Location, Interpretation and Translation

Olga Popova [1,2] and Petra Dadić [3]

[1] University of Cadiz, 9 Paseo Carlos III St., Cadiz, 11003, Spain
[2] Institute of Applied Linguistics (ILA), 16 Avenida Duque de Nájera, Cadiz, 11002, Spain
[3] University of Split, 33 Ruđera Boškovića St., Split, 21000, Croatia

Abstract

Wordplay is a vital aspect of human communication, involving creative language use to convey multiple meanings and induce humor. Automating wordplay analysis is challenging but made possible by advances in natural language processing (NLP). This study focuses on detecting, localizing, interpreting, and translating wordplay using Python and AI methods. Cultural influences on humor and wordplay are considered, particularly in English, French, and Spanish. The JOKER track at CLEF 2023 aims to advance automated humor analysis by bringing linguists, translators, and computer scientists together. Four pilot tasks are proposed: pun detection in multiple languages, pun interpretation, and pun translation from English to French and Spanish. The study provides an introduction, background on puns and wordplay, an overview of CLEF 2022 and 2023, and discusses methods, results, and future research directions. By leveraging NLP techniques, this work tries to bridge linguistic and computational approaches to enhance automated wordplay analysis.

Keywords

Pun detection, pun location, pun interpretation, pun translation, CLEF2023, JOKER, automatic humor analysis

1. Introduction

Wordplay is an essential aspect of human communication that involves the creative use of language to convey multiple meanings or to produce a humorous effect. In our daily life we resort to different language resources to express our feelings and emotions, one of these resources is wordplay. It is not necessarily humorous, it can contain irony or sarcasm, and if sometimes it is already difficult to capture it in a human communication, to do it automatically is logically even more complicated. However, it is not impossible, with the advent of natural language processing (NLP) techniques, machines can now perform these tasks with increasing accuracy and efficiency. In this working notes we focus on the detection, location, interpretation and translation of word sets using the Python programming language and different methods provided by artificial intelligence and machine learning.

We must take into account that humor and wordplay is a cultural phenomenon that is linked to the historical experience and background knowledge of the speakers of each language. Since we worked with English, French and Spanish, we had to be attentive to the particularities of each of these languages.

To advance in the automation of humor and wordplay analysis, we decided to take part in the JOKER track at CLEF 2023. The goal is to bring together linguists or translators and computer

[1]CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece
EMAIL: olga.popova@uca.es (A. 1); petradadic313@gmail.com (A. 2)
ORCID: 0000-0001-7084-3140 (A. 1)

scientists to further the computational analysis of humor. This workshop proposed three pilot tasks [1]:

- Pilot Task 1: Detection of puns in English, French, and Spanish
- Pilot Task 2: Location and interpretation of puns in English, French, and Spanish
- Pilot Task 3: Translation of puns from English to French and Spanish

These working notes are organized as follows: after the introduction, there is the background section where pun and wordplay definitions and an important terms for interpretation are described, moreover, in this section we do an overview of CLEF 2022 and CLEF 2023; the third section is dedicated to the approach (data and methods description), the fourth section is the discussion of the results, and the fifth section is the conclusions.

2. Background

In this section we will make a brief overview of the state-of-the-art. In addition, we will clarify some terms by providing their definitions and necessary explanations, such as *wordplay, pun, synonym, target synonym* and *machine translation*. We will also summarize what was done in the CLEF 2022 edition, which will help us to choose the trajectory of our work.

2.1. *Pun* and *wordplay* definitions

The first two tasks consist in wordplay detection and location. In this way, we should start defining the two main terms of this task, which are *wordplay* and *pun*.

To begin with, we turn to some of the most relevant online dictionaries: Oxford Learner's Dictionaries and Cambridge Dictionary. The definitions that can be found there are the following:

- *Wordplay1*: making jokes by using words in a clever and humorous way, especially by using a word that has two meanings, or different words that sound the same (Oxford Learner's Dictionaries).
- *Wordplay2*: the activity of joking about the meanings of words, especially in an intelligent way (Cambridge Dictionary).
- *Pun1*: the clever or humorous use of a word that has more than one meaning, or of words that have different meanings but sound the same (Oxford Learner's Dictionaries).
- *Pun2*: a humorous use of a word or phrase that has several meanings or that sounds like another word (Cambridge Dictionary).

At first glance it may seem that these two terms mean the same thing or in other words they are synonyms. But we have to say that there is a lot of research from the linguistic point of view that shows that the term *wordplay* is much broader. Winter-Froemel [2] defines the *wordplay* like: "a historically determined phenomenon in which a speaker produces an utterance - and is aware of doing so - that juxtaposes or manipulates linguistic items from one or more languages in order to surprise the hearer(s) and produce a humorous effect on them".

Moreover, Winter-Froemel [2] proposes a classification in the form of a table, which allows us to understand that the phenomenon of wordplay is much broader and more complex than in the definition of the two dictionaries mentioned at the beginning of this section. This table is reproduced below.

Table 1
Subtypes of wordplay in a large sense and verbal humour [2]

	Verbal Humour					
	Wordplay (broad sense)			…?		
Subtypes	Wordplay (narrow sense)	Soundplay	Ludic deformation	Ludic reinterpretation	Ludic innovation	Ludic translation
paradigmatic example	*(7) Less bread. No jam.*	*(11) She sells sea-shells*	*(23) Ingliche Pipole*	*(3) the rubber tears*	*(25) Nasen-fahrad*	*(32) Love song in all languages*

	(39) Ness-Café	*on the sea-shore.*				
basic level	lexical	sublexical	sublexical	lexical	conceptual	textual
textual status	local	mostly pervasive	mostly local	local	local	pervasive
basic procedure	combination / juxtaposition of preexisting (conventional) items and / or creation of new items	combination of preexisting (conventional) items	substitution of sublexical elements of conventional items; sublexical innovation	reinterpretation of ambiguous structure by H in a way not intended by S	combination; lexical innovation	substitution
formal / functional relation between units involved	homonymy / polysemy/ paronymy	paradigmatic similarity	functional similarity / equivalence of sublexical units	homonymy / polysemy	conceptual association	textual equivalence
(implicit) metalinguistic dimension	highlighting arbitrariness of language and / or motivational dimension of language	highlighting formal similarities	highlighting arbitrariness of language / linguistic convention	highlighting arbitrariness of language	highlighting motivational dimension of language	highlighting divergence of languages and varieties
(major) subcategories and related categories	wordplay in praesentia / in absentia; wordplay with or without lexical innovation	tongue twisters, lipograms, palindromes, etc.				cf. parody

This table confirms the definition of Delabastita [3]: “Wordplay is the general name for the various textual phenomena in which structural features of the language(s) used are exploited in order to bring about a *communicatively significant confrontation* of two (or more) linguistic structures with *more or less similar forms and more or less different meanings*”.

Regarding puns, Attardo [4] defines it like: “a textual occurrence in which a sequence of sounds must be interpreted with a reference to a second sequence of sounds, which may, but need not, be identical to the first sequence, for the full meaning of the text to be accessed”.

Based on all of the above, we can conclude that *wordplay* is a broader term and *pun* is part of *wordplay* as many other subtypes presented in Table 1. As Attardo [4] says “the field of wordplay is beset by terminological problems”, which will not be solved in these working notes. However, we

have to be clear that in all tasks we actually work not so much with *puns* but with *wordplay*, because this one can take a lot of forms we observe in our data.

2.2. Important terms for interpretation

One of the tasks is dedicated to the interpretation of wordplays. From the content of the training data provided by the creators of these tasks, we understood that the interpretation consisted of finding *pun synonyms* and *target synonyms* for the previously located wordplays (locations). In this section we consider it necessary to provide explanations for these two terms.

To observe the difference between *pun synonym* and *target synonym* we will take one of the examples we have in the training data.

- *Example 1*. Old chicken farmers never die, they just have a dozen aches.

In the sentence the wordplay is based on the use of words that sound similar but have different meanings.

In this case, the *pun synonym* is the word "dozen", which sounds like the word "dying." The pun works because the phrase "dozen aches" sounds like "doesn't ache" when spoken out loud. The phrase "Old chicken farmers never die" is a play on the familiar saying "Old soldiers never die, they just fade away", and the word "dozen" is used as a pun to replace the word "dying" in the original saying.

On the other hand, the *target synonym* in this sentence is the word "aches", which is the word that is the focus or target of the pun. The phrase "dozen aches" creates a play on words with the phrase "doesn't ache", which gives the impression that the old chicken farmers are not dying but instead just experiencing some minor aches and pains.

Overall, the use of *pun synonyms* and *target synonyms* in this sentence helps to create a clever and humorous expression that plays on the sound and meaning of words. By using language in this way, the sentence creates a humorous twist on a familiar saying and adds a playful tone to the discussion of aging and physical discomfort.

This way we can conclude that the *pun synonym* is a word that is used in a pun in place of another word that has a similar sound or spelling; and the *target synonym* is a word that is the focus or target of a pun or wordplay.

2.3. Humor and wordplay translation

Nowadays, translation in general and humor translation in particular are very important, especially due to globalization and intercultural communication. Translating jokes and puns is a real challenge for translators, because they are not faced with a simple search for lexical and grammatical equivalents, but must take into account the particularities of each language, interpret cultural references and make the joke funny to the recipient of the translation.

Translators can find humor in almost any type of assignment, be it a political speech, a recipe video, movie or series. The work of the translators of "The Last of Us" series recently released on HBO is truly admirable. They had to translate the jokes from a fictional book called "No Pun Intended: Volume Too", which appear throughout the series and play an important role in the plot. Such puns like “3.14% of sailors are Pi Rates”, “I used to be addicted to soap. But I'm clean now”, “You wanna hear a joke about pizza? Never mind, it was too cheesy” and many others were translated into Spanish, Italian, Portuguese, French among others. In the case of this series, the translation, at least into Spanish, is very successful. But there are many other movies or series where sometimes the translated jokes are not understood. The choice of strategy for any given pun depends on various factors [5], and while strategies that preserve wordplay are generally preferable, they are often the most challenging to pull off. If human translators have a hard time making a decision about the translation of a joke and not fail, machine translation still has a lot to improve.

Although we must say that with the advent of neural machine translation (NMT), the results have improved significantly. As Stahberg [6] says, “the advent of NMT certainly marksone of the major milestones in the history of MT, and has led to a radical and sudden departure of mainstream research from many previous research lines.” NMT has already been widely adopted in industry [7, 8, 9, 10] and is deployed in production systems by Google, Microsoft, Facebook, Amazon, SDL,

Yandex, and many more. In this work we have used both neural machine translation (GPT-3, Bloom, Google) and some pre-training models (Simple-T5 and EasyNMT).

Researchers have been taking an increasing interest in the use of language technology in creative translation in general, and humor translation in particular, including the integration of MT systems into human translation workflows [11, 12, 13, 14]. However, puns are not suitable for off-the-shelf, end-to-end MT systems, particularly those based on the prevailing neural paradigm [15]. And while others have pointed out the potentials of digital tools to assist literary translation processes [16], no currently available tool specifically supports the translation of puns [17].

However, in this work we try to approach the translation of puns using different MT tools, assess the quality of the translations made and analyze the main problems. All this is described in parts 3 and 4.

2.4. Overview of CLEF 2022 and CLEF 2023

The JOKER project aimed to advance the automation of creative-language translation by organizing the JOKER track at CLEF 2022. Participants succeeded in wordplay location, but the interpretation tasks raised difficulties and the binary classes were unbalanced. Style shift in the translation of puns could pose an issue. In the previous edition of CLEF, Pilot Task 1 involved classifying and explaining instances of wordplay, with one team successfully predicting the location and interpretation of the wordplay. Pilot Task 2 required participants to translate single terms containing wordplay, with all participants successfully translating all instances. In Task 3, participants had to translate phrases containing wordplay from English to French, but only 13% of automatically translated wordplays were successful [18]. These results lead us to the conclusion that machine translation is still inadequate for translating puns. Successful machine translations were apparently accidental due to the existence of the same ambiguous word in both languages.

In relation to CLEF 2023, the results of the previous edition and the lessons learned from it have been taken into account. JOKER-2023 aims to expand tasks including Spanish, simplify shared tasks and focus on one type of wordplay, puns. Puns are often considered untranslatable, making them a good focus for the research. As we have already mentioned, the three shared tasks for JOKER-2023 are: detection of puns, location and interpretation of puns, and translation of puns from English to French and Spanish. The hope is that with a larger data set and more interconnected tasks, JOKER-2023 will provide better performance [1].

The following is a brief description of the tasks and the data provided for their completion.

The first task is to detect puns in English, French, and Spanish. Pun detection involves distinguishing between texts containing a pun and those that do not. The data is split into training and test sets. The English data includes positive examples from SemEval-2017 Task 7 and SemEval-2021 Task 12, while the French data is based on a corpus created in 2022 and will be improved and extended for JOKER-2023. The Spanish data set is collected from various web sources. Evaluation will be done using precision, recall, accuracy, and F-score measures for pun detection, and precision, recall, and F-score measures for pun location [1].

Pun Location and Interpretation is the second task where systems have to identify the words with double meanings in a pun-filled text and find the two meanings of a pun. The data sets will contain synonyms or hypernyms of the words involved in the pun, except for those that share a spelling with the pun. This annotation scheme allows systems to avoid relying on a specific sense inventory or notation scheme. The data for this task will be taken from Task 1, with each pun word annotated with two sets of words, one for each meaning of the pun. The evaluation will be based on precision, recall, and F-score metrics used in word sense disambiguation, with each instance being scored as the average score for each of its senses [1].

The objective of the third task is to translate English puns into French and Spanish while preserving the original wordplay using the PUN→PUN strategy. Updated training and test sets of punning jokes in English-French will be provided, as well as new sets in English-Spanish. The evaluation of the translations will be done manually by trained experts who will evaluate features such as preservation of lexical field, sense, wordplay form, style shift, and humorousness shift, as well as the presence of errors in syntax, word choice, and other factors. The runs will be ranked based on the

number of successful translations that maintain the form and meaning of the original wordplay, and we will also experiment with other semi-automatic metrics [1].

In the next section, 3. Approach, we will describe the methods used to carry out each of the tasks, which is why this section is divided into three sections (by tasks), every section includes data description and one more section with methods used to execute the tasks.

3. Approach

In this section we will give a brief description of the data provided to perform each task and the methods used to solve them.

3.1. Task 1: Detection of puns in English, French, and Spanish

The data for Task 1 (pun detection) consists of positive examples, which are short jokes containing a single pun. These examples will be drawn from existing corpora and new collections. Negative examples, used only for the pun detection subtask, will be generated through data augmentation techniques. These techniques involve manually or semi-automatically editing positive examples to remove the wordplay while preserving most of the remaining meaning. This approach aims to minimize differences in length, vocabulary, style, etc., to prevent neural approaches from simply detecting these differences.The train and the test data are provided in JSON and CSV formats [1].

Below is a table with the size of the data provided.

Table 2
Data size for task 1.1 (pun detection)

	EN	FR	ES
train data (jokes)	5292	3998	839
test data (jokes)	8474	16871	4263

3.2. Task 2: Location and interpretation of puns in English, French, and Spanish

Then in task 2.1 (pun location) we have to identify a specific word that contains the pun. We have been given the training data with the word already found and the test data only with the text of the joke. The output (results) is practically the same as in task 1.1, but instead of yes or no we would have to put the location.

The data for task 2.2 (pun interpretation) is based on the positive examples, where the pun word is annotated with two sets of words representing each sense of the pun from task 1. Each set will include synonyms or hypernyms of the respective sense or, in the case of heterographic puns, the underlying target word [1].

Table 3
Data size for task 2.1 (pun location)

	EN	FR	ES
train data (jokes)	2874	2000	439
test data (jokes)	3519	6654	1835

Table 4
Data size for task 2.2 (pun interpretation)

	EN
train data (jokes)	2874
test data (jokes)	8474

3.3. Task 3: Translation of puns in English, French, and Spanish

The objective of this task is to translate English punning jokes into French and Spanish while preserving both the form and meaning of the original wordplay. The translation approach should follow Delabastita's pun→pun strategy, as described in the typology of pun translation strategies. For instance, the English example "I used to be a banker but I lost interest" could be translated into French as "J'ai été banquier mais j'en ai perdu tout l'intérêt," where the pun is maintained due to the shared ambiguity between "interest" and "intérêt." [1]

The task will provide an updated training and test set of English-to-French translations of punning jokes, as well as new sets of English-to-Spanish translations, similar to the English-to-French datasets created for JOKER-2022. The train and the test data are provided in JSON and CSV formats.

Table 5
Data size for task 3 (pun translation)

	EN-FR	EN-ES
train data (jokes)	5837	564
test data (jokes)	5726	5726

3.4. Methods

In this section we will describe how we approach each of the tasks. First, we provide a table summarizing the methods used for each exercise.

Table 6
Methods per tasks

Task 1 EN / ES / FR	Task 2.1 EN / ES / FR	Task 2.1 EN / ES / FR	Task 3 EN-ES / EN-FR
TF-IDF	SpaCy	GPT3	SimpleT5
SimpleT5	SimpleT5	BLOOM	GPT3
Random	GPT3	SpaCy	Googletrans
Naive Bayes	BLOOM	SimpleT5	EasyNMT-Opus
MLP		WordNet	EasyNMT-mbart
Logistic Regression			BLOOM
Fast Text			

3.4.1. TF-IDF Ridge

TF-IDF (term frequency-inverse document frequency) is a numerical statistic used in information retrieval to measure the significance of a word in a document within a collection or corpus. It combines the frequency of a term in a document (TF) with its rarity across the corpus (IDF) to determine its importance [19]. It was used for task 1. We used TF-IDF Ridge with the following code:

```
from sklearn.linear_model import RidgeClassifier
clf = RidgeClassifier(tol=1e-2, solver="sparse_cg")
clf.fit(X_train, y_train)
pred = clf.predict(X_test)
```

3.4.2. SimpleT5

SimpleT5 is a Python framework that is open-source and built on top of PyTorch-lightning and Transformers. It simplifies the process of training and fine-tuning T5 models. With SimpleT5, you can easily train T5 models for various NLP tasks like summarization, translation, question-answering, and

text generation. It provides a streamlined and user-friendly interface, allowing you to quickly develop and deploy T5 models for your specific NLP needs [20].

Since this method has a very wide scope of operation, in other words, it can be trained for almost any task, so we used it in all exercises. Also, the function for all tasks was the same, the only thing we had to do was to change the names of the columns in the training data: `source_text` and `target_text`.

3.4.3. Random

As task 1.1. was all we had to do was to get the answer "YES" or "NO", one of the methods we used was Random. Specifically, we turned to the randit() function.

```
data_name["Random"]=["YES" if randint(0,1)==1 else "NO" for i in range(len(data_name))]
```

`randint()` is a built-in function of the random module in Python that returns a random integer between the higher and lower limit passed as parameters. `randint()` takes only integer type parameters and generates an integer type random value.

3.4.4. Naive Bayes

The multinomial Naive Bayes classifier is suitable for classification with discrete features, such as word counts in text classification. Typically, integer feature counts are required for the multinomial distribution. However, in practice, fractional counts like tf-idf can also be used. This classifier utilizes Bayes' theorem to calculate the probability of a document belonging to a specific class based on the frequencies or counts of its features. It is widely used in natural language processing tasks due to its simplicity and efficiency in text classification [21]. It was used for task 1 with vectorised text sentences.

3.4.5. MLP

The MLP (Multi-Layer Perceptron) Classifier is a type of artificial neural network that is commonly used for classification tasks. The MLP Classifier learns from labeled training data to make predictions on unseen data by adjusting the weights of the connections between nodes through a process called backpropagation. It is a versatile classifier capable of handling complex patterns and is widely used in various domains, including image recognition, natural language processing, and recommendation systems [22]. We used it in task 1.1 with vectorised text sentences as what we needed was precisely the classification.

3.4.6. Logistic regression

Logistic regression is a widely used classification technique belonging to the group of linear classifiers. It shares similarities with polynomial and linear regression. It is a fast and straightforward method, making it convenient for result interpretation. While primarily used for binary classification, logistic regression can also be extended to handle multiclass problems. Its simplicity and interpretability make it a fundamental tool in the field of classification [23]. It also was used to solve task 1 with vectorised text sentences.

3.4.7. Fast Text

We also use FastText for the detection of puns. It is a text classifier. Text classification is a task that involves assigning documents, such as emails, posts, or product reviews, to specific categories or tags. These categories can represent various aspects, such as sentiment, topic, or language. Machine learning is the predominant approach used to develop text classifiers, where classification rules are learned from labeled data. So we pre-trained the model and applied it to our test data in task 1 with vectorised text sentences.

3.4.8. SpaCy

SpaCy is an open-source software library for advanced natural language processing, written in the programming languages Python and Cython [24]. In the case of our work we try to apply this method to the location (task 2.1) of puns and interpretation or more specifically synonym search (task 2.2).

3.4.9. GPT3

GPT is a pre-trained transformer large scale language learning model [25]. GPT3 was used for task 2.1, 2.2, and 3, limiting the number of examples (100 per task) due to the token issue.

3.4.10. BLOOM

BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) is a transformer-based large language model [26]. BLOOM was used for task 2.1, 2.2, and 3, limiting the number of examples (100 per task) due to the token issue.

The prompts used for each task are provided below.

Task 2.1 (EN), prompt for GPT3 and BLOOM:

```
"Sentence: Herbivores come in browns and graze.
Pun: graze
Sentence: I used to do rock climbing as a youth, but I was much boulder back then.
Pun: boulder
Sentence: She dumped him because of all their lousy dates. After all, whining and dining does get tiresome after a while.
Pun: whining
Sentence: When you're wearing a watch on an airplane, time flies.
Pun:"
```

Task 2.1 (FR), prompt for GPT3 and BLOOM:

```
"Sentence: Certaines personnes prennent des photos magnifiques et les coupent en morceaux. C'est un vrai puzzle pour moi.
Pun: puzzle
Sentence: Docteur, docteur, je continue à penser que je suis une cuillère. - Assieds-toi là et ne remue pas. Suivant.
Pun: remue
Sentence: Le mannequin qui avait rejoint les forces de l'air était une bombe.
Pun: bombe
Sentence: Ce n'était pas la pomme poussant sur l'arbre de la connaissance le problème, c'était les deux poires en dessous.
Pun:"
```

Task 2.1 (ES), prompt for GPT3 and BLOOM:

```
"Sentence: Los diabéticos no deberían tener dulces sueños.
Pun: dulces
Sentence: Al amanecer me van a pasar por la guillotina y mi mujer ya ha firmado la separación.
Pun: separación
Sentence: Me mudé y tuve que buscar otros médicos después de estar cinco años con el mismo quiropráctico. Fue un mero ajuste.
Pun: ajuste
Sentence: Un científico estaba haciendo un gran experimento con químicos en estado líquido cuando se cayó y pasó a ser parte de la solución.
Pun:"
```

Task 2.2, prompt for GPT3 and BLOOM:

```
“Pun: conviction
Pun synonyms\/hypernyms: article of faith;strong belief
Pun: graze
Pun synonyms\/hypernyms: conviction
Pun: reproved
Pun synonyms\/hypernyms: admonish;reprove;reproof
Pun: boulder
Pun synonyms\/hypernyms:”
```

Task 3 (EN-FR), prompt for GPT3: `"Translate this from English into French:\n\n".`

Task 3 (EN-FR), prompt for BLOOM:

```
“Original: Save the whales, spouted Tom.
Translation: "Sauvez les baleines", proclama Tom à tout évent.
Original: A skier retired because he was going downhill.
Translation: Le skieur est parti à la retraite. Il n'arrivait pas à remonter la pente.
Original: My wife uses a kitchen implement to shred garlic and parmesan cheese, which I hate. It really is the grater of two evils.
Translation: Ma femme écoute du hip hop quand elle cuisine des carottes ou du gruyère. Je n'aime pas ça mais elle me dit que ça l'aide à râper.
Original: Staying at the trendy, new hotel was the inn thing to do.
Translation: Je rêvais de dormir dans cet hôtel.
Original: The fireplaces of oriental doctors have an Asian flue.
Translation:”
```

Task 3 (EN-ES), prompt for GPT3: `"Translate this from English into Spanish:\n\n".`

Task 3 (EN-ES), prompt for BLOOM:

```
"Original: Diabetics should not be allowed to have sweet dreams.
Translation: Los diabéticos no deberían tener dulces sueños.
Original: I'm going to the guillotine at dawn and my wife has already collected my severance pay.
Translation: Al amanecer me van a pasar por la guillotina y mi mujer ya ha firmado la separación.
Original: After 5 years with the same chiropractor, I moved and had to change doctors. It was quite an adjustment.
Translation: Me mudé y tuve que buscar otros médicos después de estar cinco años con el mismo quiropráctico. Fue un mero ajuste.
Original: A scientist doing a large experiment with liquid chemicals was trying to solve a problem when he fell in and became part of the solution.
Translation: Un científico que hacía un gran experimento con productos químicos líquidos estaba intentando solucionar un problema cuando cayó en que él se convertiría en parte de la solución.
Original: Old electricians never die, they just keep plugging away.
```

```
Translation:"
```

3.4.11. WordNet

WordNet [27] is a large English language dictionary that can be used in python as a part of NLTKlibrary [28]. We used it in task 2.2 (interpretation) to search synonyms.

3.4.12. Googletrans

Googletrans is a free and unlimited python library that implemented Google Translate API [29]. It was used to solve task 3.

3.4.13. EasyNMT-Opus and EasyNMT-mbart

EasyNMT is an easy to use python library for Machine Translation. This library is developed by NLP researchers from UKP Lab, TU Darmstadt [30]. It has a lot of models, but we decided to use two more well-known ones (Opus and mbart) for the pun translation task.

4. Results

In this section we will present the main results of our approach applied to the tasks and their discussion.

4.1. Task 1. Pun detection

For the results of our runs for pun detection we compare four parameters: precision (proportion of positive actually correct identifications), recall (proportion of actual positives was identified correctly), f1 (harmonic mean of the precision and recall), and accuracy (fraction of predictions the model got right) in English, Spanish and French data.

Table 7
Precision for task 1 (pun detection)

	EN	FR	ES
Random	0.2554194156456173	0.421484695672569	0.4205729166666667
FastText	0.2562081198265668	0.552456286427976	0.4075342465753425
NB	0.2612369043595809	0.567320703653585	0.4769094138543517
LogisticRegression	0.2614403600900225	0.58439664600802	0.5
TF-IDF	0.2690937870993272	0.587731811697574	0.5334143377885784
MLP	0.2778568041725936	0.564996614759647	0.5545335085413929
SimpleT5	0.319792158715163	0.612199693303799	0.4431017119838872

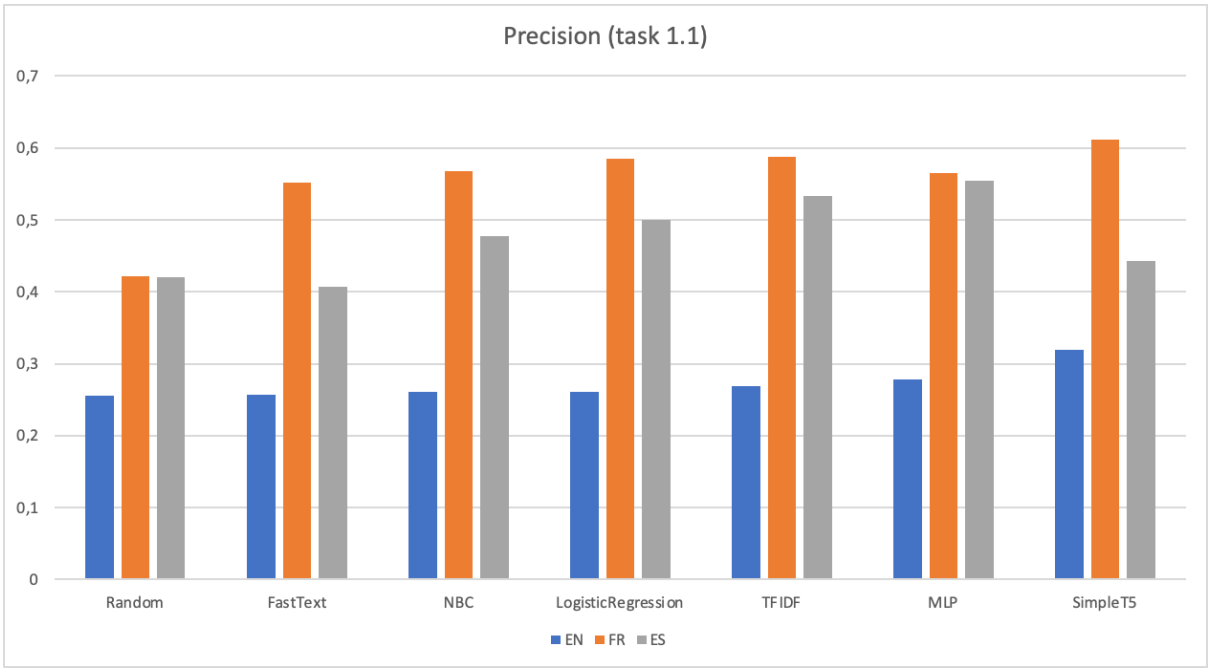

Figure 1: Precision for task 1 (pun detection)

We can observe that the best precision in general was achieved for the French data, then for Spanish data, and the precision for English data is less than 50%. The method that worked best with the French and English data is SimpleT5 and MLP for Spanish.

Table 8
Recall for task 1 (pun detection)

	EN	FR	ES
FastText	0.803461063040791	0.625	0.25
LogisticRegression	0.861557478368356	0.490546218487394	0.603993971363978
MLP	0.724351050679851	0.443277310924369	0.628862094951017
NB	0.955500618046971	0.5640756302521	0.631876412961567
Random	0.669962917181705	0.678571428571428	0.677091183119819
SimpleT5	0.836835599505562	0.462184873949579	0.676902788244159
TF-IDF	0.840543881334981	0.461134453781512	0.620949510173323

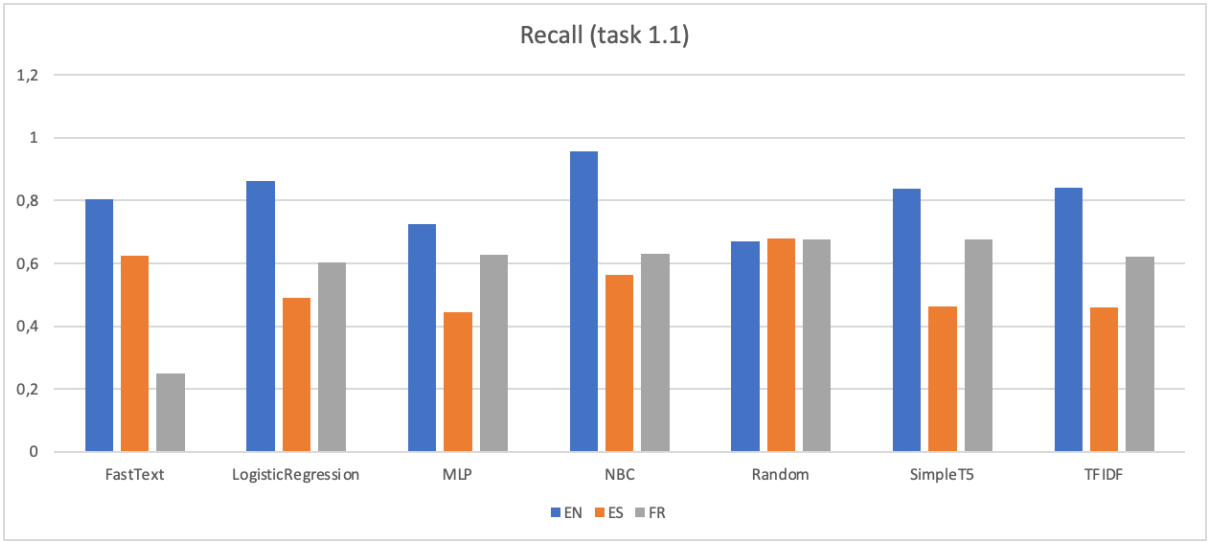

Figure 2: Recall for task 1 (pun detection)

As for recall metrics, the runs for English data have the best results, especially the NBC method (95%). The results for Spanish and French data are practically the same. We can make a curious observation that the metric for the data in the three languages is practically the same with Random (66-67%). Thus the probability of success with this method is logically more or less 50%, but it is not reliable, as the predictions are made randomly.

Table 9
F1 for task 1 (pun detection)

	EN	FR	ES
FastText	0,3885236103	0,4933665008	0,344228275
LogisticRegression	0,4011510791	0,4952279958	0,5940337224
MLP	0,4016449623	0,4927028605	0,5952211127
NB	0,4102972399	0,5168431184	0,5978609626
Random	0,3698396452	0,5192926045	0,5195518612
SimpleT5	0,4627477785	0,4524421594	0,6429274403
TF-IDF	0,4076738609	0,4946478873	0,6038842067

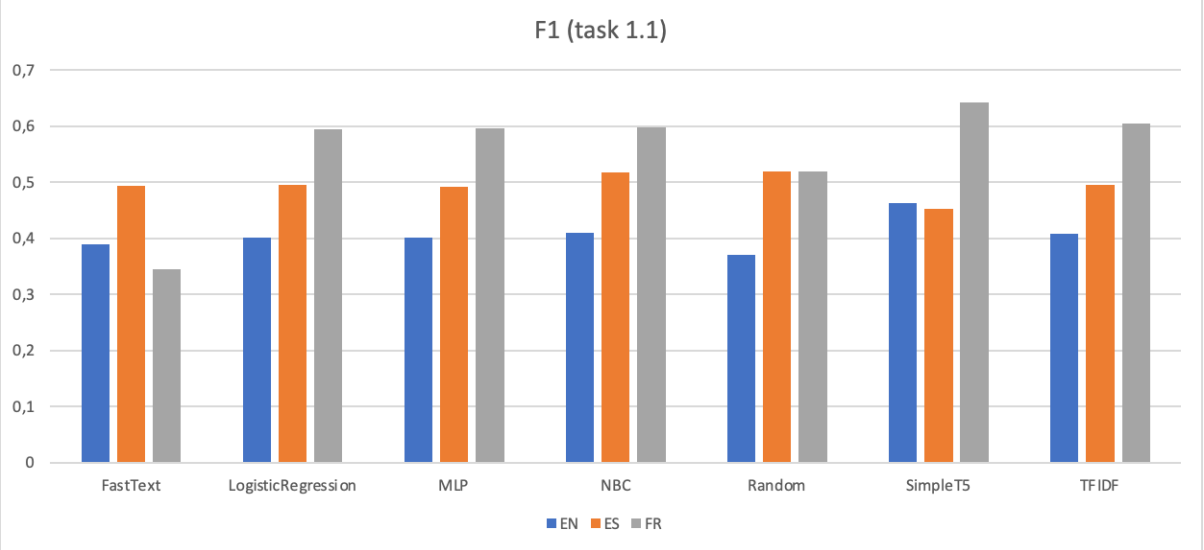

Figure 3: F1 for task 1 (pun detection)

As the f1 metric is a harmonic mean of precision and recall.

Table 10
Accuracy for task 1 (pun detection)

	EN	FR	ES
FastText	0.3572101791	0.4547077198	0.6072399596
LogisticRegression	0.3462142633	0.5751896475	0.6595976074
MLP	0.451460886	0.6122266845	0.6473238561
NB	0.301916431	0.5519857207	0.6494989513
Random	0.4197298146	0.4663096832	0.4836479453
SimpleT5	0.5061262959	0.5247657296	0.6899712577
TF-IDF	0.3792020107	0.5997322624	0.6641031617

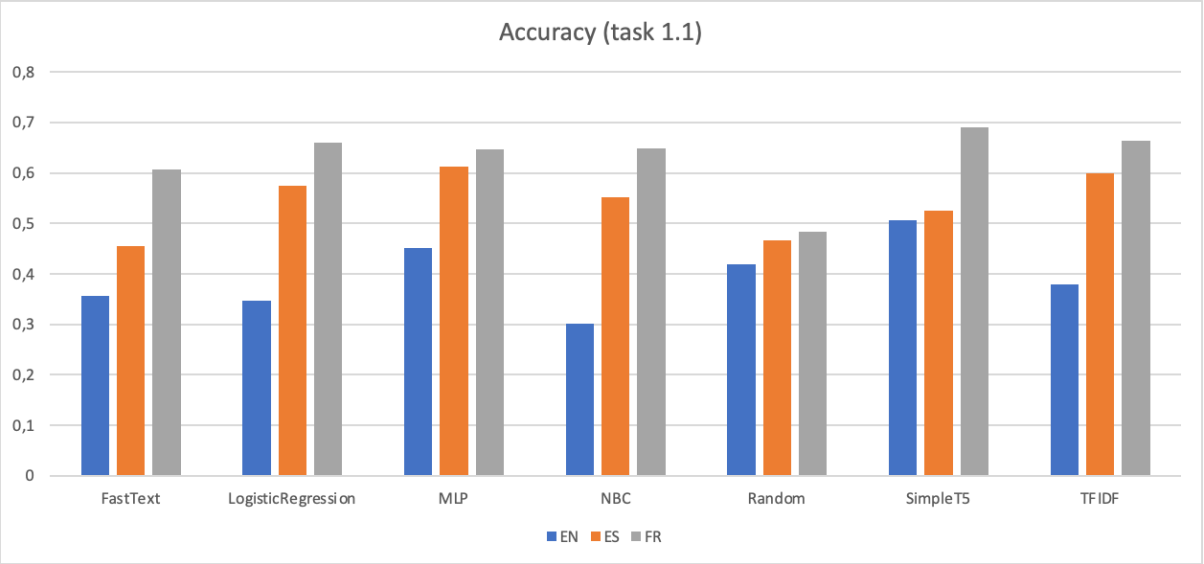

Figure 4: Accuracy for task 1 (pun detection)

Finally, we will analyze the accuracy. It is surprising that the results for the English data are worse than for the French and Spanish data, when most of the methods and libraries are supposedly developed on the basis of English. Due to the figure 4 we can conclude that the best method for pun detection in English is SimpleT5 (50% accuracy), in French is SimpleT5 too but with better score (69% accuracy), and for Spanish is MLP method (61% accuracy).

It is really interesting to note that each method has given such different results for each language. But it must also be said that none of the methods applied has reached even 90% accuracy. So the results in general, even if they are above 50%, can be improved, since it is simply a binary prediction. We consider that with the current level of development of Artificial Intelligence the methods applied should work better with this type of exercise.

4.2. Task 2.1. Pun location

Since this task does not involve any binary classification, but rather the location of the pun, we can analyze only the accuracy, comparing the results for English, Spanish and French.

Table 11
Accuracy for SimpleT5 and SpaCy (Task 2.1. Pun location)

	EN	FR	ES
SimpleT5	0.7950207469	0.4543501611	0.828125
SpaCy	0.444813278	0.0	0.2416666667

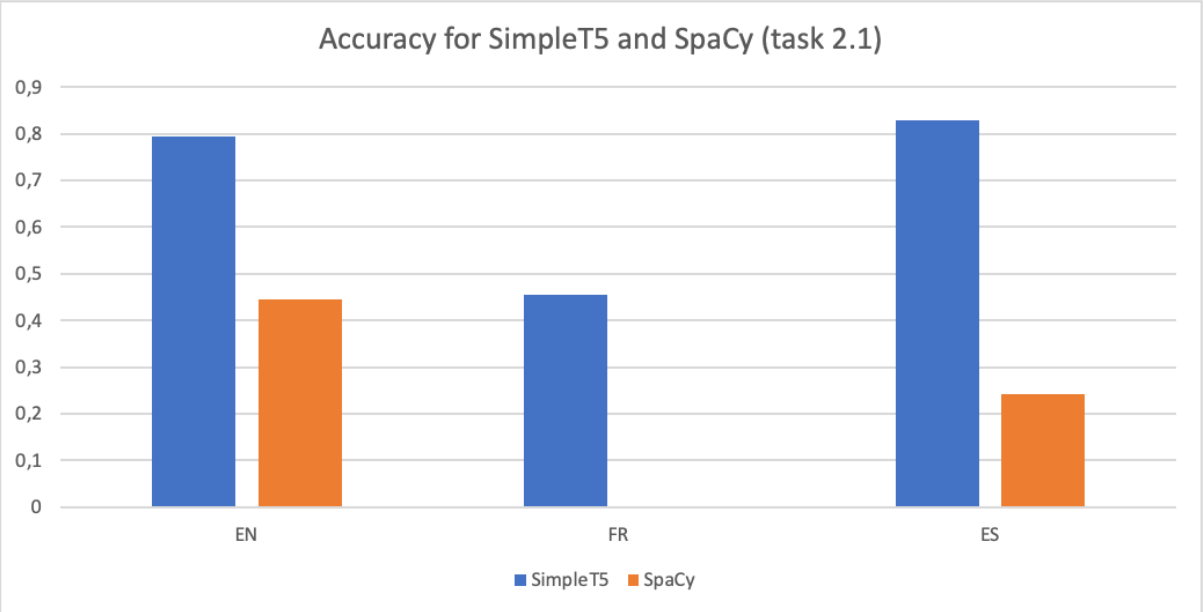

Figure 5: Accuracy for SimpleT5 and SpaCy (Task 2.1. Pun location)

Table 12
Accuracy for BLOOM and GPT3 (Task 2.1. Pun location)

	EN	FR	ES
BLOOM	0.65625	0.3384615385	0.4385964912
GPT3	0.8125	0.4615384615	0.8771929825

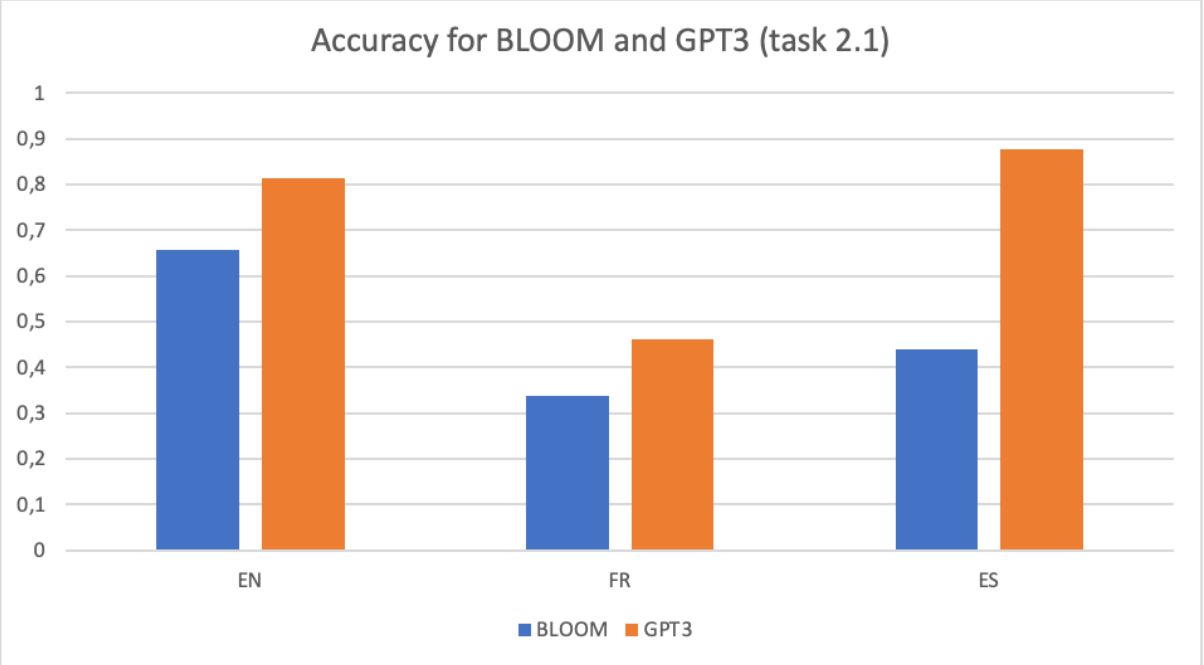

Figure 6: Accuracy for BLOOM and GPT3 (Task 2.1. Pun location)

According to figure 6 the best results for the pun location were achieved with GPT3 (81% for English, 46% for French and 87% for Spanish). Then we can observe that SimpleT5 was very successful (79% for English, 45% for French and 82% for Spanish). Despite our expectations, BLOOM has not been very accurate. And SpaCy (figure 5) directly proved that it does not have such a well-developed library, especially in French, as its result was 0%. In general, the results for French

data are worse than for English or Spanish, we can suppose that the French database is not well-developed and the methods we use are not well-trained for this language.

4.3. Task 2.2. Pun interpretation

We will now briefly discuss the results of the second task. First, we provide a table with some examples of pun interpretation (source synonyms and target synonyms). In this table we provide only some of the examples.

Table 13
Pun interpretation results

Method	Text	Location	Source synonym	Target synonym
BLOOM	Soft drink inventors saw a big popportunity.	popportunity	chance; opportunity	drink
	The unveiling of the statue was a monumental occasion.	monumental	large; massive; stately	statue
SimpleT5, WordNet	OLD PHILOSOPHERS never die, they just retire to their own premises.	premises	premises, premise, premiss, assumption, premise	premises, premise, premiss, assumption, premise
SpaCy, WordNet	OLD GEOMETRY TEACHERS never die, they just go off on a tangent.	tangent	tangent, tangent, tan	tangent, tangent, tan
BLOOM, WordNet	''I prefer trout to salmon,'' Tom said officiously.	trout	trout, trout	trout, trout
GPT3, WordNet	I never found sending signals from ships challenging. I always had a flare for it.	flare	flare, flair, flare, flare, flash	flare, flair, flare, flare, flash
GPT3	I never found sending signals from ships challenging. I always had a flare for it.	flare	blaze; flame; ignite; flare up	grey hairs; grey hair; gray hair; gray hairs
	OLD TELEPHONE OPERATORS never die, they just become disconnected.	disconnected	separated; isolated; divided; broken	eggs

	When his wife asked for wooden walls in the basement, they had a panel discussion.	panel	board; committee; group; assembly	eggs

As we can see only BLOOM gives the consistent results, differentiating source and target synonyms. The other methods still need a lot of training in order to achieve good interpretation results, as many of them give the same words for two columns. We expected better results from GPT3, but possibly due to some mistake or an incorrect prompt in many of the cases it has given the word "eggs" as target synonym regardless of the joke and the pun.

4.4. Task 3. Pun translation

In this subsection we will describe the results of Task 3 (pun translation). As indicated above, we use six different methods to carry out this task. Below, we present a table containing some examples of the translations (English-Spanish) carried out with each of the methods and ordered in order from the best to the worst translation. It should be noted that the spelling and grammar obtained are preserved in the table (we emphasize this, as SimpleT5, for example, contains numerous spelling mistakes in Spanish). In this table we provide only some of the examples.

Table 14
Pun translation (EN-ES) results

Priority	Method	Source text	Target text
1	GPT3	When the fog burns off it won't be mist.	Cuando la niebla se queme, ya no será niebla.
		The boy swallowed a pillow, the hospital described his condition as comfortable.	El niño tragó una almohada, el hospital describió su condición como cómoda.
		''Eating uranium makes me feel funny,'' said Tom radiantly.	"Comer uranio me hace sentir raro", dijo Tom radiante.
2	BLOOM	When the fog burns off it won't be mist.	Cuando se disipe la niebla no será una bruma.
		The boy swallowed a pillow, the hospital described his condition as comfortable.	El niño se tragó un cojín, el hospital describió su estado como cómodo.
		''Eating uranium makes me feel funny,'' said Tom radiantly.	Tom dijo radiantemente que

			comer uranio le hacía sentirse raro.
3	Googletrans	I've got to fix the automobile, said Tom mechanically.	Tengo que arreglar el automóvil, dijo Tom mecánicamente.
		Those with scissors shouldn't use cutting words.	Aquellos con tijeras no deben usar palabras de corte.
		A skier retired because he was going downhill.	Un esquiador se retiró porque iba cuesta abajo.
4	EasyNMT-Opus	I've got to fix the automobile, said Tom mechanically.	Tengo que arreglar el auto, dijo Tom mecánicamente.
		Those with scissors shouldn't use cutting words.	Aquellos con tijeras no deberían usar palabras cortantes.
		A skier retired because he was going downhill.	Un esquiador se retiró porque iba cuesta abajo.
5	EasyNMT-mbart	I've got to fix the automobile, said Tom mechanically.	He de reparar el automóvil, dijo Tom de forma mecánica.
		Those with scissors shouldn't use cutting words.	Los que tienen cuerdas no deberían usar las palabras de corte.
		A skier retired because he was going downhill.	Un esquiador se retiró porque estaba descendiendo.
6	SimpleT5	I've got to fix the automobile, said Tom mechanically.	El automático tena que repararlo, dijo Tom mecanicamente.
		Those with scissors shouldn't use cutting words.	Cuando tenan córculos no deberan utilizar trabajadores.

		A skier retired because he was going downhill.	Un skier se retragó porque estaba descendido.

As we can see, the translation done by SimpleT5 is quite incoherent and contains many spelling mistakes. Not even the general meaning of the sentence is translated well, not to mention the puns. We believe that this problem is due to very little training data.

The other methods produced fairly acceptable translations. Jokes where the wordplay is quite obvious and easy, e.g. “I've got to fix the automobile, said Tom mechanically” or “The boy swallowed a pillow, the hospital described his condition as comfortable” are understood in the target language. However, the methods used were less successful with the more complicated jokes to convey their meaning in Spanish. The best translations of more complex jokes were achieved with GPT3 and BLOOM. Among the examples in the table is the following: “When the fog burns off it won't be mist”. Only BLOOM has managed to retain the Spanish pun and produce a comic effect. The translation was as follows: “Cuando se disipe la niebla no será una bruma”. To translate the word "mist", BLOOM is the only method that chooses "bruma", having many other simpler options. This word is similar to "broma", only one letter is changed, so the pun is achieved. We don't know if it is a matter of luck or a "conscious" choice of translation.

We obtained very similar results for the English-French translation, the only difference being that SimpleT5 performed better (the training data for the French was much more extensive.).

Table 15
SimpleT5 EN-FR translation

Priority	Method	Source text	Target text
6	SimpleT5	I've got to fix the automobile, said Tom mechanically.	Je dois réparer l'automobile, dit Tom mécaniquement.
		Those with scissors shouldn't use cutting words.	Ceux qui ont des ciseaux ne devraient pas utiliser des mots coupants.
		A skier retired because he was going downhill.	Un skieur a pris sa retraite parce qu'il allait descendre.

We can conclude that the translation of puns still poses a great challenge for machine translation. We believe it needs more training materials and time to improve the results obtained. Very often even humans are not able to convey a joke in another language as it was said in the source language, as it is not always possible to find the direct equivalents of the puns. Therefore, translators have to make many transformations and sometimes even completely rewrite the joke, which requires more intellectual, cultural and linguistic effort than artificial intelligence has today.

4.5. Limitations

The tasks were solved by two students, i.e. not professional programmers or computer scientists, but beginners in the field. Thus, during the implementation of the activities, some problems and lack of knowledge inevitably arose.

In addition, we highlighted as a limitation the use of the free Google Colaboratory, as the execution of some methods was very time-consuming. We found that Google Colab stops execution after 4 hours. Therefore, in the case of translation with Googletrans we had to split the data into four parts to perform the task. For SimpleT5 we had to use the GPU-connection, but it is not permitted to

use more than one notebook with GPU at the same time. Moreover, sometimes Google Colab had a restriction to run with GPU more than some times in one day.

5. Conclusions

In this report we have described some theoretical issues about humor, puns and wordplay. And we have provided the results of three tasks: pun detection, location, interpretation and translation. We have submitted 21 runs for pun detection, 12 runs for pun location, 6 runs for pun interpretation and 12 runs for pun translation carried out with different methods such as TF-IDF, SimpleT5, Random, Naive Bayes, MLP, Logistic Regression, Fast Text, SpaCy, GPT3, BLOOM, WordNet, Googletrans, EasyNMT-Opus and EasyNMT-mbart. So we had both pre-training methods and neural models. We worked with the data in English, French and Spanish.

We compared the results obtained with all these methods and came to a general conclusion that artificial intelligence can perform the more or less simple tasks such as pun detection and location with relative success, but more complicated tasks such as pun interpretation and translation still require a lot of improvement. So, does artificial intelligence have a sense of humor? So far not much, but he is developing it little by little and still has a long way to go to get the perfect results, which we also believe will not be possible without human review as well. At least not in the very near future.

As perspectives for future work we could focus on tasks 2 and 3 to see how it would be possible to teach the machine to differentiate between source and target synonyms and to find more creative ways of translation. Also, more attention needs to be paid to the development of libraries, models and databases not only in English but also in other languages.

6. References

[1] L. Ermakova, T. Miller, A. G. Bosser, V. M. Palma Preciado, G. Sidorov, and A. Jatowt, Overview of JOKER - CLEF-2023 track on Automatic Wordplay Analysis, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, A. Giachanou, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Fourteenth International Conference of the CLEF Association (CLEF 2023), 2023.

[2] E. Winter-Froemel, Approaching Wordplay, in: E. Winter-Froemel (Ed.), The Dynamics of Wordplay, volume 3 of Crossing Languages to Play with Words, De Gruyter, Berlin/Boston, 2016, pp. 11-46.

[3] D. Delabastita (Ed.), Wordplay and Translation, The Translator / Studies in Intercultural Communication, volume 2, number 2, Routledge, London and New York, 1996.

[4] S. Attardo, Universals and puns in humorous way, in: E. Winter-Froemel and V. Thaler (Eds.), Cultures and Traditions of Wordplay and Wordplay Research, volume 6 of The Dynamics of Wordplay, De Gruyter, Berlin/Boston, 2018, pp. 89-109. doi: 10.1515/9783110586374-005.

[5] I. Klitgard, "Wordplay and Translation", in: K. Malmkjaer (Ed.), The Routledge Handbook of Translation and Linguistics, Routledge, New York, 2018, pp. 233-248. doi: 10.4324/9781315692845-16.

[6] F. Stahlberg, "Neural Machine Translation: A Review", Journal of Artificial Intelligence Research 69, 2020, pp. 343-418. doi: 10.1613/jair.1.12007

[7] Y. Wu, M.Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey., et al., Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv preprint: arXiv:1609.08144, 2016.

[8] J. Crego, J. Kim, G. Klein, A. Rebollo, K. Yang, J. Senellart, E. Akhanov, P. Brunelle, A. Coquard, Y. Deng, et al., SYSTRAN's pure neural machine translation systems, arXiv preprint arXiv: 1610.05540, 2016.

[9] T. Schmidt & L. Marg, How to move to neural machine translation for enterprise-scale programs - an early adoption case study, 2018.

[10] P. Levin, N. Dhanuka, T. Khalil, F. Kovalev & M. Khailov, Toward a full-scale neural machine translation in production: the booking.com use case, arXiv preprint arXiv: 1709.05820, 2017.

[11] J. Moorkens, A. Toral, Sh. Castilho, and A. Way, “Translator’s Perceptions of Literary Post-Editing Using Statistical and Neural Machine Translation”, Translation Spaces 7 (2), 2018, pp. 240-262.
[12] A. Toral and A. Way, “What Level of Quality Can Neural Machine Translation Attain on Literary Text?”, in: J. Moorkers, Sh. Castilho, F. Gaspari and S. Doherty (Eds.), Translation Quality Assessment: From Principles to Practice, Cham, Springer, 2018, pp. 263-287.
[13] K. Taivalkoski-Shilov, “Ethical Issues Regarding Machine(-Assisted) Translation of Literary Texts”, Perspectives 27 (5), 2019, pp. 689-703.
[14] M. A. Jiménez-Crespo, “The ‘Technological Turn’ in Translation Studies: Are We There Yet? A Transversal Cross-Disciplinary Approach”, Translation Spaces 9 (2), 2020, pp. 314-341. doi: 10.1075/ts.19012.jim.
[15] T. Miller, “The Punster’s Amanuensis: The Proper Place of Humans and Machines in the Translation of Wordplay”, in: Proceedings of the Second Workshop on Human-Informed Translation and Interpreting Technology, Incoma, Shoumen, 2019, pp. 57-64. doi: 10.26615/issn.2683-0078.2019_007.
[16] Y. Roy, Using Computers in the Translation of Literary Style: Challenges and Opportunities, Routledge, New York, 2019. doi: 10.4324/9780429030345.
[17] K. Waltraud and T. Miller, “Human-computer interaction in pun translation”, in: J. L. Hadley, K. Taivalkoski-Shilov, C. S. C. Teixeira, and A. Toral (Eds.), Using Technologies for Creative-Text Translation, Routledge, New York, 2022, pp. 66-88. doi: 10.4324/9781003094159-4.
[18] L. Ermakova, T. Miller, F. Regattin, A. G. Bosser, C. Borg, E. Mathurin, G. Le Corre, S. Araújo, R. Hannachi, J. Boccou, A. Digue, A. Damoy and B. Jeanjean, Overview of JOKER@CLEF 2022: Automatic Wordplay and Humor Translation Workshop, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction: 13th International Conference of the CLEF Association, CLEF 2022, Bologna, Italy, September 5-8, 2022, Proceedings, Sprinter-Verlag, Berlin, Heidelberg, 2022, pp. 447-469. doi: 10.1007/978-3-031-13643-6_27 .
[19] A. Rajaman and J. D. Ullman, “Data Mining”, Mining of Massive Datasets, 2011, pp. 1-17. doi: 10.1017/CBO9781139058452.002.
[20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21, 2020, pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html
[21] A. McCallum, Andrew, "Graphical Models, Lecture2: Bayesian Network Representation" (PDF). Archived (PDF) from the original on 2022-10-09, 2019. URL: https://people.cs.umass.edu/~mccallum/courses/gm2011/02-bn-rep.pdf.
[22] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY, 2009.
[23] M. Stojiljkovic, Logistic Regression in Python, URL: https://realpython.com/logistic-regression-python/#logistic-regression-in-python.
[24] M. Hannibal, "Introducing spaCy", URL: https://explosion.ai/blog/introducing-spacy.
[25] R. Dale, Gpt-3: What’s it good for?, Natural Language Engineering 27, 2021, pp. 113-118. doi:10.1017/S1351324920000601.
[26] BigScience Workshop, BLOOM (revision 4ab0472), 2022. URL: https://huggingface.co/bigscience/bloom. doi:10.57967/hf/0003.
[27] C. Fellbaum, WordNet: An Electronic Lexical Database, Bradford Books, 1998. URL: https://mitpress.mit.edu/9780262561167/.
[28] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, ” O’Reilly Media, Inc.”, 2009.
[29] Googletrans 3.0.0, URL: https://pypi.org/project/googletrans/.
[30] K. Subramanyam Kalyan, Neural Machine Translation using EasyNMT Library, 2022, URL: https://mr-nlp.github.io/posts/2022/07/mt-easynmt/.