applied sciences

MDPI

*Article*

# Applying Automatic Translation for Optical Music Recognition's Encoding Step

**Antonio Ríos-Vila [1], Miquel Esplà-Gomis [1], David Rizo [1,2], Pedro J. Ponce de León [1] and José M. Iñesta [1,*]**

[1] Department of Software and Computing Systems, University of Alicante, Ctra. San Vicente del Raspeig s/n, 03690 Alicante, Spain; arios@dlsi.ua.es (A.R.-V.); mespla@dlsi.ua.es (M.E.-G.); drizo@dlsi.ua.es (D.R.); pierre@dlsi.ua.es (P.J.P.d.L.)
[2] Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana, 03011 Alicante, Spain
* Correspondence: inesta@dlsi.ua.es

**Abstract:** Optical music recognition is a research field whose efforts have been mainly focused, due to the difficulties involved in its processes, on document and image recognition. However, there is a final step after the recognition phase that has not been properly addressed or discussed, and which is relevant to obtaining a standard digital score from the recognition process: the step of encoding data into a standard file format. In this paper, we address this task by proposing and evaluating the feasibility of using machine translation techniques, using statistical approaches and neural systems, to automatically convert the results of graphical encoding recognition into a standard semantic format, which can be exported as a digital score. We also discuss the implications, challenges and details to be taken into account when applying machine translation techniques to music languages, which are very different from natural human languages. This needs to be addressed prior to performing experiments and has not been reported in previous works. We also describe and detail experimental results, and conclude that applying machine translation techniques is a suitable solution for this task, as they have proven to obtain robust results.

**Keywords:** optical music recognition; machine translation; machine learning; computer vision; intermediate representation; humdrum



**Citation:** Rios-Vila, A.; Esplà-Gomis, M.; Rizo, D.; Ponce de León, P.J.; Iñesta, J.M. Applying Automatic Translation for Optical Music Recognition's Encoding Step. *Appl. Sci.* **2021**, *11*, 3890. https://doi.org/11093890

Academic Editor: Seungmin Rho

Received: 12 March 2021
Accepted: 20 April 2021
Published: 25 April 2021



## 1. Introduction

As a part of human cultural heritage, musical compositions have been transmitted over the centuries. One of the means of preserving and transmitting such compositions is by visually encoding them in documents called music scores. A significant proportion of all music that has been written throughout history is only available via physical documents that have never been stored in a structured digital format that allows their indexing, retrieval and digital processing. Given the cost of manual transcription, automatic processing would be preferable, and could be done in the same way as transcribing text from images of documents.

Optical music recognition, usually referred to as OMR, is a field of research that investigates how to computationally read music notation in documents [1]. Traditionally, OMR has been addressed differently from tasks that might seem similar, such as optical character recognition (OCR). For instance, most of the existing literature on OMR is framed within a multi-stage workflow, with steps involving image binarization and staff-line detection and removal [2,3]; symbol classification [4,5]; notation assembly [6,7]; and semantic encoding, this last step being the subject of interest of this work.

Recent advances in machine learning, specifically deep learning, have changed the way we approach OMR. Instead of using legacy pipelines, based on statistical and data-driven approaches, we leverage the capabilities of neural methods, which merge some of these stages and present better and more straightforward implementations.

One of these advances is the end-to-end recognition of symbols within a staff. Specifically, we found in the literature one approach based on convolutional recurrent neural networks (CRNN) trained using the connectionist temporal classification (CTC) loss function [8]. It deals with a staff-region input image and retrieves a sequence of graphical symbols. It presents them in a so-called agnostic representation, which describes the recognized tokens by their visual features (shape and vertical position in the staff). This format does not take into account any semantics or musical meaning, since it only describes those features that are visible to the human eye, not the interpretations that they may have within their context.

As we can deduce, this OMR-based output makes it difficult for this technology to translate its research results into practical applications, since they are not in a standard format that can be used in any notation editor or digital musicology framework, such as Music21 [9] or Verovio Humdrum Viewer (VHV) [10], which prevents OMR from being a turnkey system that the user can interact with. To do this, a final step has to be performed: encoding. This is the process that converts graphics-based data into a standard music notation output, which can then be used in other applications.

There exists little literature that addresses this step, by using rule-based systems or grammar [6,11]. However, these solutions present scalability issues, as new rules must be manually introduced if the source or target encodings are updated, making it difficult to maintain the system as music encoding methods multiply. Recent work on the topic has shown that the application of machine learning techniques, specifically machine translation (MT) approaches, can be a viable solution to solve this problem without scale issues, as the encoding step closely resembles the challenges that machine translation scenarios have faced. We explored this topic in a previous work with a simple end-to-end model. In [12] we used nonstandard simplified semantic encoding to explore the feasibility of applying these techniques to the encoding step. However, that work only explored the surface of the issue.

In the present work, we dive into this exploration and evaluate the application of MT techniques to solve the OMR encoding problem. Our main goal with this article is to outline the fundamental ideas to take into account when performing a machine translation task between digital music encodings, which has not been discussed in previous works. We also applied said ideas in this particular step, while evaluating several MT techniques applied to standard semantic encodings, and observed their performances.

The rest of the article is organized as it follows: In Section 2, the methodology used in this research is presented, where we present and justify our output semantic encoding of choice, and specify the approaches for the automatic translation between the agnostic and semantic encodings: the statistic approach and the neural approach. In Section 3, an experimental setup is established where the corpora used in these experiments are presented, and describe the specific challenges and corner-cases that involve using this kind of automatic translation system for different music encoding representations. In Section 4, results are presented and the performance of the proposed models is discussed in-depth. Finally, in Section 5, we conclude this work and explain our final thoughts about the convenience of using these MT techniques for the encoding step in the OMR pipeline.

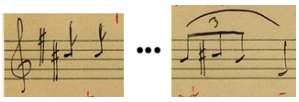
## 2. Methodology

Several approaches to obtain a digital score file from an image have been proposed so far [1]. When leaving aside any considerations of image processing and document analysis, and focusing on how the final standard notation encoding is obtained, two main approaches can be found: an end-to-end, image-to-semantics approach; and systems that produce an intermediate representation that needs to be converted into a standard format, such as the Music Encondig Initiative (MEI) [13,14], MusicXML [15] and **kern* [16]. In the first approach, a semantic format is obtained directly from an input image [17]. It has been shown that the main drawback of these methods is that they are very sensitive to distortion in input images. In the second approach, an intermediate representation that tries to explain the content of the score in terms of graphical elements is used. However,

this graphics-based encoding needs to be converted to a standard music notation encoding. Meaningful examples of this category include that of Pacha et al. [18], where a notation graph is generated from a set of primitives, such as note heads, stems, beams and accidentals. Additionally, in [17], a simpler representation denoted as agnostic is used that consists of musical symbol sequences instead of considering their constitutive primitives. This approach has been shown to be more robust than the image-to-semantic one.

This work focuses on the second phase of this latter approach: the translation between the agnostic encoding of single monodic staves into a standard semantic format, which is an extension of the ***kern* encoding described and discussed below in Section 2.1. We expect that the lessons learned from this work can be adapted to other intermediate to semantic conversion approaches.

Specifically, a sequence of symbols is extracted from the segmented image (Figure 1a), using any method. These are agnostic symbols characterized by a graphic glyph, such as accidental, rest, clef or note, and their vertical positions on the staff (see Figure 1b), and they do not have a direct and unambiguous semantic meaning. For instance, in the excerpt from Figure 1a, the second sharp is not assigned either the function of key signature, or an alteration of the next eighth note in the staff. They are just a sequence of two pitch alteration symbols detected at a particular vertical position, in this case the fifth line and the third space. The digit "3" found in the second part of the example image is labeled as a digit "3", and its actual meaning (it may be a numerator in a fractional time signature, or a triplet number in a tuplet) will be assigned later when this representation is translated to semantic coding. The reader can find more detailed information about this agnostic encoding in [17,19].



(**a**) Excerpt of a common western music notation manuscript

```
clef.G2:L2, accidental.sharp:L5, accidental.sharp:S3,
note.eighth_up:S3, note.eighth_up:S3, ...
slur.start:S5, note.beamedRight1_up:L3, accidental.sharp:S3,
digit.3:S5, note.beamedBoth1_up:S3, note.beamedLeft1_up:L3,
slur.end:S5, note.quarter_up:S1
```

(**b**) Agnostic encoding of the symbols found in the excerpt

```
**kern
*clefG2
*k[f#]
8cc#/
8cc#/
...
(12bL/
12cc/#
12bJ/)
4f#/
*-
```

(**c**) Translation to semantic encoding

**Figure 1.** An example of the translation of the agnostic encoding corresponding to a common western music manuscript excerpt shown in (**a**) to a semantic encoding, see their representations in (**b**) and (**c**). Note how the meaning of the two accidentals after the clef is disambiguated into a G major key and an accidental of the first eighth note.

Therefore, the task under analysis is the translation of this sequence of graphical symbols into a semantic encoding expressed as a standard music representation encoding.

The **kern encoding is used in the example in Figure 1c and an extension of it will be considered as the output language for the approaches described below in Sections 2.2 and 2.3.

### 2.1. Output Semantic Encoding

An important determinant of any translation effort is to find a semantic output encoding to be obtained by the translation system. To achieve this goal, it is necessary to analyze which coding is most beneficial in terms of exportability and subsequent compatibility with the aforementioned musicological tools. The first options that can be considered are the most widespread semantic encoding methods in libraries and musicology contexts: MEI and MusicXML, which represent the components and metadata of a musical score in an XML-based markup attribute encoding. Despite being extended encodings, these semantic representations have a major drawback when considering their use in a machine translation system, as they are very verbose. That is, the amount of tokens required to represent a musical symbol is greater than in the case of agnostic encoding. This also means that the target language vocabulary would be greater than the input one. This is not convenient, since in this situation word alignment between encodings is more difficult to perform by machine learning systems, and would result in a more erratic predictive model.

In previous research and experiments carried out on this subject [20], a semantic encoding based on Humdrum **kern was proposed. This is a robust and widespread standard used in musicological projects. The benefits mentioned in that work also apply to this case, which can be summarized in its simple vocabulary, its compatibility with dedicated music software [10] and its possibility of being converted directly into the aforementioned encodings, which is its most remarkable and celebrated benefit. However, as it is explained in the mentioned publication, Humdrum **kern has some drawbacks when representing specific music symbols, so an extension to it is proposed, namely, **kern$^*$, which is able to be directly converted and reverted from/to **kern. Since some of our corpora were used in that earlier work and the advantages of the encoding remain the same, we decided to use this language as the output semantic encoding for our experimentation.

### 2.2. The Statistical Approach

This approach is based on the family of techniques known as statistical machine translation (SMT) [21]. SMT is a data-driven approach to MT in which several independent models are combined to produce a translation from a text in the source language (SL) into the target language (TL). The basic models in an SMT system are:

- The translation model: this model is learned from a parallel corpus, i.e., a set of sequences in the SL with their translations in the TL, and is used to obtain, given a phrase (i.e., a word $n$-gram) in the SL, the most probable equivalent phrases in the TL.
- The language model: this model is automatically learned from a TL monolingual corpus, and is used to obtain the probability that a predicted segment belongs to the TL.

In addition to the translation model and the language model, modern SMT approaches integrate additional models, such as a distortion model (to reordering words during translation) or a length penalty. All these models are integrated through a log-linear combination; the weight of each model in this combination is set by a simplex-type algorithm that optimizes an automatic quality score on a development set.

The combination of all these models allows SMT systems to provide balanced translations in terms of translation accuracy and TL fluency. However, it should be noted that the context available during translation is determined by the sizes of the sentences and the order of the language model. This is a clear disadvantage compared to neural approaches that deal with context in a more sophisticated way (see Section 2.3). Therefore, we hypothesize that, when translating from the agnostic encoding of a musical work that includes a clef other than the most repeated one, or having a key signature with a number of flats or sharps, SMT methods may lack information to correctly handle the pitch of notes that are not near the beginning of the staff. In order to confirm this hypothesis, we have created

a modified version of the agnostic representation, named contextual agnostic encoding. This new notation extends the standard agnostic notation by concatenating to each symbol the clef and key signature context. The clef is encoded with a suffix equal to the agnostic clef sign. The key signature is encoded as the number of contiguous sharps or flats it is made of (see Figure 2).

```
clef.G:L2, accidental.flat:L3, accidental.flat:S4, accidental.flat:S2,
digit.2:L4, digit.4:L2, note.eighth_down@clef.G:L2@3b:L3, verticalLine:L1,
note.eighth_up@clef.G:L2@3b:L2, note.sixteenth_up@clef.G:L2@3b:L2,
note.sixteenth_up@clef.G:L2@3b:L2, note.eighth_up@clef.G:L2@3b:S2,
note.eighth_up@clef.G:L2@3b:S2, verticalLine:L1, note.quarter_down@clef.G:L2@3b:L3,
note.eighth_down@clef.G:L2@3b:L3, note.eighth_down@clef.G:L2@3b:S3, verticalLine:L1,
note.eighth_down@clef.G:L2@3b:L4, note.eighth_down@clef.G:L2@3b:S3,
note.eighth_down@clef.G:L2@3b:L4, note.eighth_down@clef.G:L2@3b:S3
```

**Figure 2.** An example of an agnostic encoding sequence including context information (see the full example in Appendix A Figure A1). Symbols representing notes are extended to include information about the respective clefs and key signatures. The string token `@clef.G:L2@3b` inserted in these symbols represents a G clef an a three flat signature (as found at the beginning of the sequence).

### *2.3. Neural Approaches*

In this part, the proposed neural models with which to approach the translation task are described. They are alternative solutions to the statistical model described above—namely, the recurrent neural network-based translation system and the transformer-based one.

#### 2.3.1. Sequence to Sequence with Attention Mechanisms

The first neural approach used in this work is based on the sequence to sequence (Seq2Seq) system proposed in [22]. This network's architecture tries to maximize the conditional probability of a given token, which in our case is a ***kern*word, based on its inputs and the previous predicted tokens. The proposed model to solve this problem was designed with a recurrent neural network (RNN), which output a prediction based on the received inputs and the previous states of the neurons. The structure of the Seq2Seq model is composed of two parts: an encoder that processes the input sequence element by element and embeds its relevant information into an internal state, often referred to as a context vector; and a decoder, which combines the internal state of the encoder and the previous predicted token to generate its equivalent embedding in the target encoding that is then used to predict the next token. Since its inception, this model has evolved in the context of machine translation. In particular, this evolution has been directed towards the attention mechanisms, which are masking matrix operations whose goal is to assist with the model's predictions by enhancing the values of the most relevant information in an input. In particular, we resort to the "Global Attention" strategy proposed by Luong et al. [23], with a scoring method given by the scalar product between the encoder and the decoder outputs. The attention matrix produced in this method acts as a weighting element in order to select which tokens are relevant to produce the next token in the target sequence. A representation of the implemented model in this work is depicted in Figure 3.

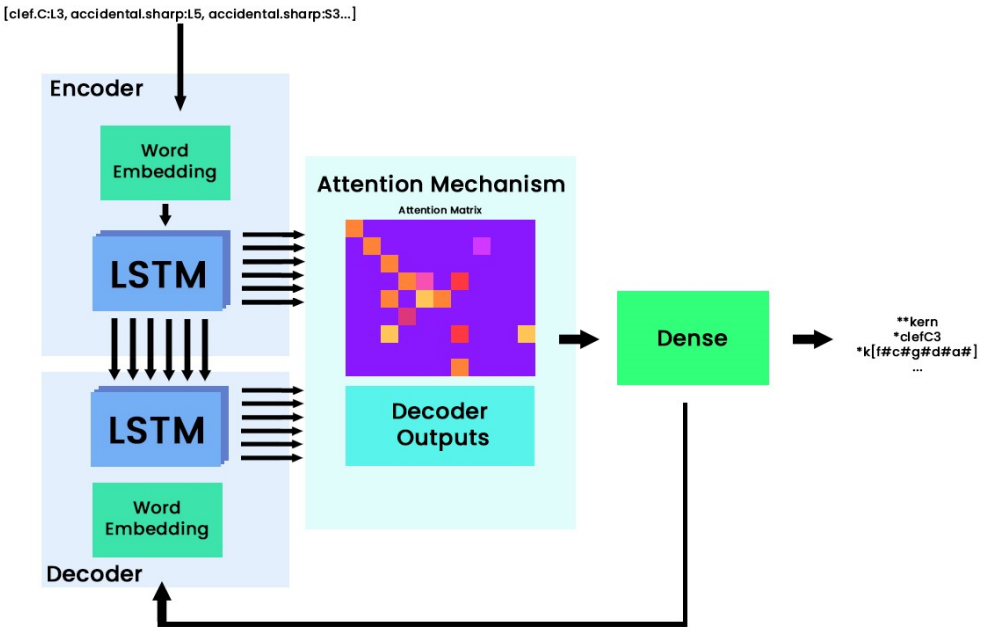



**Figure 3.** Proposed neural architecture for the Seq2Seq with attention mechanisms model. In this schema, the LSTM (long short-term memory layers) boxes represent the recurrent layers implemented in the encoding, and decoding steps and the dense box represents the last linear layer which classifies the next token in the target sequence. The output of this layer is then propagated to the decoder in order to provide this new information to predict the next token in the target sequence.

#### 2.3.2. The Transformer

The Seq2Seq model described above is a reference model in machine translation for performing the kinds of tasks we face. Due to the significant improvements over traditional models that the attention-based approaches bring, they have gained significant popularity during recent years. This has lead the research community to develop new systems that take advantage of these elements. This leads us to the second model used for this work: the transformer, which currently represents the state-of-the-art in natural language processing tasks [24,25]. The transformer, firstly presented in [26], is a model that proposes a new approach to computing the encoding and decoding process presented in the previous subsection. Instead of implementing recurrent neurons to compute this process, this model presents an alternative methodology based on linear neurons and operations of the attention mechanism, which are referred to in the literature as the multi-headed attention neurons, which outperformed the original recurrent systems, among other benefits, such as computing benchmarks.

A simple description for this mechanism is that the system evaluates the relevance of the sequence elements to describe a specific token (self-attention mechanism). This is applied to the encoding and decoding steps. By replacing the recurrent layers with the multi-headed attention (MHA) mechanisms, the system, as reported in [26], gains significant performance without loosing the benefits that the state-of-the-art recurrent neurons, such as long short-term memory (LSTM) ones [27], have. The improvements in the training benchmarks the transformer has in comparison to the recurrent models are also remarkable, since it is an easier model to parallelize, as it is not bounded to the concept of sequentiality that the recurrent networks have. However, his lack of sequentiality is something that has to be addressed, since in our case it is mandatory to have a certain order for the tokens in sequence. However, the original paper also proposes an elegant word coding named positional encoding, which consists of mathematical functions that produce singular results depending on the position of the token in the sequence. Thanks to this process, which is also accompanied by masking operations to avoid positional prediction, the transformer is

able to process text sequences without loosing positional information, making it a powerful model for use in machine translation applications.

In this work, we implemented a simple version of the transformer model, as the most refined versions require large amounts of annotated data, which were not available here. An image representation of the proposed architecture is described in Figure 4.

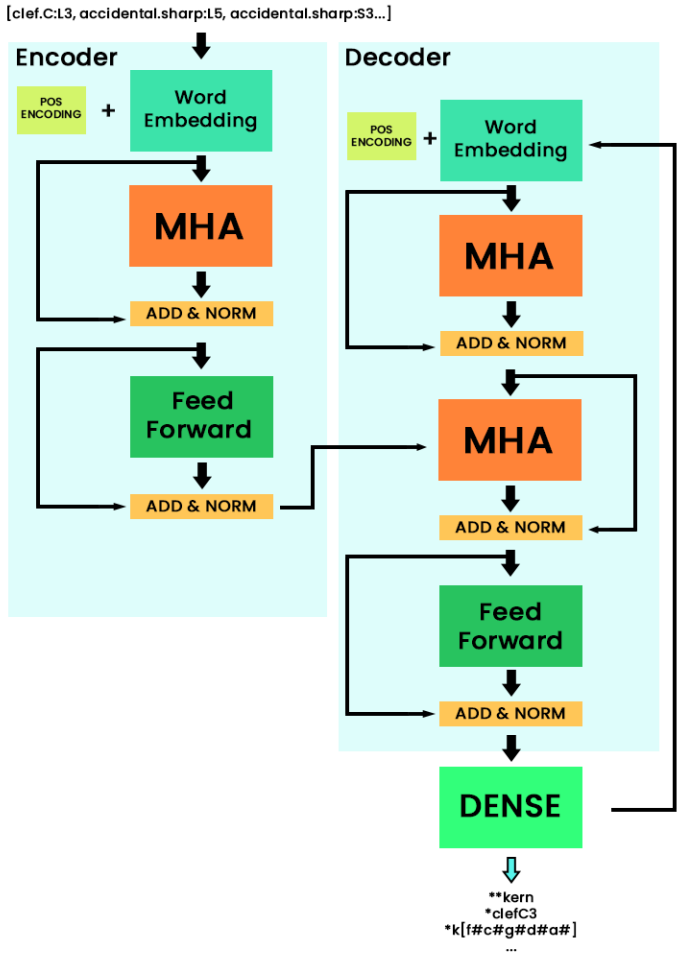



**Figure 4.** The proposed transformer neural architecture, which was adapted from [26]. The MHA blocks are the multi-headed attention mechanisms described in this article and in the reference paper, and the POS ENCODING blocks refer to the mentioned positional encoding functions which are computed with the respective inputs.

## 3. Experimental Setup

This section explains all the relevant aspects considered during the experimental phase. First, the corpora used are presented and described. Then, some relevant challenges to be taken into account when performing machine translation between musical encodings and the ambiguous cases that may hinder this process are discussed. Finally, the particular implementations of the models used are detailed.

### *3.1. Corpora*

In order to understand the behavior of the different translation approaches, three different datasets of dissimilar nature have been used. In all cases, they consisted of sets of monodic staves, either in mensural or modern notation. For both types of notation, the same agnostic symbol dictionary was used. However, the target semantic encoding was different according to the type of notation: in the case of mensural notation ***mens* [28], a variant of standard ***kern* was used. Modern notation was encoded using ***kern*$^{*}$.

The first corpus, named *Zaragoza*, is a compilation of 17th and 18th century manuscripts from the "Cathedral of Our Lady of the Pillar" in Zaragoza (Spain) [29]. This dataset is an evolution of the dataset "Zaragoza" created manually and introduced in [30] and refined using the MuRET tool [31]. A complete sample is shown in Appendix A, Figure A2.

The second corpus, *Camera-PrIMuS*, is described in [17]. This synthetic dataset was obtained from the RISM collection [32] and automatically rendered to images, and transcoded to agnostic and semantic notation. The rendering process introduces a variety of image distortion operators in order to produce samples close to actual scanned or photographed printed scores. The original version of the corpus contains 87,678 samples, each containing a single staff. In this work, samples containing mensural notation have been removed from this corpus, in order to have one type of notation per corpus. Thus, the filtered dataset actually contained 87,316 staves (see a complete example in Appendix A Figure A3).

The third corpus, named FMT, is introduced in this work. It is a collection of four groups of handwritten score sheets of popular Spanish songs taken from the "Fondo de Música Tradicional IMF-CSIC" [33]. It is a set of melodies transcribed by musicologists between 1944 and 1960. All music sheets contain monodic staves written in modern notation. The agnostic encodings have been obtained by the same procedure applied in Camera-PrIMuS. See Figure A1 for a sample of this corpus.

Some of the music sheets in FMT have a particular feature: in order to save space for the actual notes in a staff, the clefs and key signatures in some of these manuscripts were written only at the beginning of the pieces, unlike the standard western common practice. This means that, in this case, most staves of a piece (except for the first) have no clef and key signature (see an example in Appendix A Figure A1). This was a common practice in the mid-20th century era for some types of printed music, at least in Spain, such as music for marching bands. This was mainly due to the fact that these manuscripts were often written on small pieces of paper, to make them manageable in different performance situations. Thus, musicians had to learn to "remember the current key" from staff to staff. Since in this experimental setup the input instances are isolated staves extracted from those manuscripts, we should expect the performance of the models in these samples to degrade due to this problem.

For the purpose of analyzing translation errors, the FMT corpus was divided into two datasets, FMT-M, which contains the pieces that have the clef and clef signature correctly annotated; and FMT-C, containing those pieces with missing key signature and clefs at the beginning of all staves except for the one at the top of each piece.

Table 1 presents the overall statistics of these corpora in terms of instances and symbols. Table 2 presents statistics about the semantic content of the corpora, including some figures about clefs and key signatures. In particular, the figures about staves with key signatures report the percentage of staves that are part of a piece with an explicitly written key signature. The average ratio of notes altered by key reports the percentage of notes on those staves whose pitch is affected by the key signature.

**Table 1.** Characterisation of datasets.

| | Zaragoza | FMT | Camera-PrIMuS |
|---|---|---|---|
| Number of staves | 140 | 872 | 87,316 |
| Mean length agnostic | 42 | 19 | 27 |
| Mean length semantic | 36 | 19 | 25 |
| Agnostic/semantic length ratio | 0.9 | 0.97 | 0.92 |
| Agnostic vocabulary size | 200 | 266 | 862 |
| Contextual agnostic vocabulary size | 506 | 432 | 13,721 |
| Semantic vocabulary size | 421 | 206 | 1,421 |
| Ratio of unknown agnostic symbols (*) | 0.09 | 0.13 | 0.00 |
| Ratio of unknown contextual agnostic symbols (*) | 0.37 | 0.28 | 0.02 |

(*) in test sets.

**Table 2.** Some statistics about the semantic contents of datasets. Numbers in the form $a \pm s(m)$ express average, standard deviation, and mode, respectively. (**) These figures include staves with absent keys, for which the keys of the first staves in the same pieces are assumed.

| | **Zaragoza** | **FMT-M** | **FMT-C** | **Camera-PrIMuS** |
|---|---|---|---|---|
| Score pages | 24 | 161 | 81 | N/A |
| Staves per page | $6.0 \pm 3.0$ (5) | $3.5 \pm 1.0$ (3) | $3.9 \pm 1.7$ (3) | 1 |
| Notes per staff | $28.5 \pm 7.3$ (28) | $12.6 \pm 5.2$ (14) | $10.6 \pm 4.1$ (10) | $15.5 \pm 5.2$ (15) |
| Other symbols per staff | $13.8 \pm 4.0$ (14) | $8.7 \pm 3.5$ (6) | $6.2 \pm 2.8$ (4) | $11.3 \pm 4.1$ (9) |
| Staves with clef | 96.5% | 100% | 25.9% | 100% |
| Staves with key signature | 81.8% | 36.7% | 52.1 (**)% | 76.7% |
| Avg. ratio of notes altered by key (*) | 13.0% | 9.5% | 24.7 (**)% | 27.2% |

(*) when key is present.

It is worth noting that the size of the contextual agnostic vocabulary can be, at the most, $15 \times 7$ times the non-contextual vocabulary size, as there are 15 possible different key signatures and up to 7 different clefs. Actually, in our corpora, the largest increase in the vocabulary size, by factor of around $\times 16$, is in the Camera-PrIMuS corpus. This poses a potential challenge for statistical models when the size of the corpus is relatively small compared to the vocabulary size, as is the case for the FMT and Zaragoza corpora.

### *3.2. Translating Monodic Staves*

In general, agnostic and semantic symbols convey more than one token of information. Most agnostic symbols have two components: the type of symbol and its position in the staff. For example, as depicted in Figure 5, the G-clef is represented by `clef.G:L2`, where `clef.G` represents the G-clef graphic glyph, and `L2` represents its position in the staff (actually, the second line (`L2`) of the staff). In the case of ***kern** or ***mens*, our choice for semantic encoding, this symbol also conveys two tokens of information, such as `clefG` for the symbol and `2` for its position. Symbols representing notes have several components as well: in the agnostic representation, `note.eighth_down:S3` represents an `eighth` note glyph with its stem pointing `down`, and its head position placed in the third space (`S3`) of the staff. In ***kern**, `8cc#` represents a C#5 (`cc#`) eighth note (`8`).

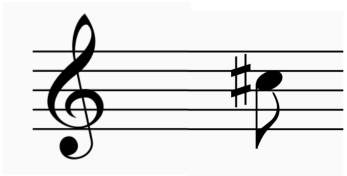



**Figure 5.** An example of an agnostic encoding and its translation into a ***kern** semantic encoding.

Note that, from the point of view of machine learning models, `note.eighth_down:S3` is a different symbol from `note.eighth_down:L3`, as `8cc#` is a different symbol from `8cc`. Additionally, note that, when translating from agnostic to semantic encoding, often two or more input symbols have to be combined into one semantic symbol, as in the case of the C# note in Figure 5.

#### 3.2.1. Translation Categories

The process of translating from a graphics-oriented representation (agnostic) to a performance-oriented (semantic) representation of music presents some specific challenges for machine learning models. Given the nature of the music data in our corpora (monodic staves), several aspects of the agnostic notation have been identified as potentially difficult to learn for translation to a semantic representation. We call them translation categories,

and we are interested in investigating which of these categories are most difficult to learn by the proposed models.

For the purpose of our research, four translation categories have been identified, in which at least two agnostic symbols found in the input sequences have to be combined to produce one semantic symbol, or decorated with some additional semantic tokens. These categories are described below. Note that input symbols to be combined do not necessarily have to be adjacent.

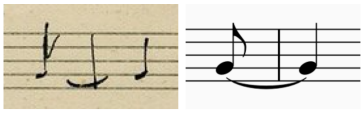
- **Slurs and ties**. Agnostic encoding of slurs and ties share the same structure: the feature is encoded by a starting symbol, followed by some continuity symbols, and ended by a ending symbol. Slurs and ties decorate the semantic representation of notes enclosed by them. In the agnostic encoding, slurs and ties are represented by the same agnostic symbol token (`slur`), as they are visually indistinguishable.
  Depending on the relative positions of these glyphs in the original score, the order in which they appear in the agnostic encoding, as a result of optical recognition, may vary. For example, two tied notes can be encoded as

  ```
  note.eighth_up:L2, slur.start:L2, slur.end:L2, note.quarter_up:L2
  ```

  or as

  ```
  slur.start:L2, note.eighth_up:L2, note.quarter_up:L2, slur.end:L2
  ```

  Figure 6 presents a example of this translation category.



```
note.eighth_up:L2, slur.start:L2, verticalLine:L1, slur.end:L2, note.quarter_up:L2

                                  [8g/  =  4g]/
```

**Figure 6.** Examples of the translation of slurs and ties. Both original manuscript snapshots and their printed counterparts are shown next to each other. The agnostic encoding and its translation to ***kern** are shown below the snapshots. This example shows how the semantic symbols encoding these notes are decorated by adding open and closing brackets to represent the tie.

- **Dotted notes**. Dots affect the duration of the note preceding them. They are encoded as separate symbols in the agnostic encoding, but have to be translated into single output symbols, as in the following example:

  ```
  note.quarter_up:S2, dot:S2  -->  4.a/
  ```

- **Pitch**. The pitch of a note in a semantic encoding depends both on the current clef and its vertical position in the staff, which usually appears at the beginning of the staff, and the vertical position of the note glyph, encoded as part of the symbol representing that note. As an example,

  ```
  clef.G:L2, ..., note.eighth_down:L3  -->  8b\
  ```

  models must learn to combine both the G clef in the second line and the eight note in the third line, which may be separated by an undefined number of other symbols, to produce a single semantic symbol (an eighth B note, represented as "`8b\`" in this case), which contains the correct token of pitch information, which is "`b`" in this example, provided by the distant clef.

- **Pitch altered by key signature**. The decoration of a note in the semantic encoding regarding including accidentals depends both on the key signature, which is found immediately after the clef in the input sequences, and on the vertical position of this notes in the staff. Again, as in the case of pitch, these symbols can be arbitrarily far from each other on the staff. Moreover, the key signature is encoded in the input as a sequence of one or more accidental symbols. For example, given

  ```
  clef.G:L2, accidental.sharp:L5, ... note.eighth_down:S1, ...
  ```

  the eighth note in the first space of the staff (S1) should be semantically encoded as an F# ('`8f#\`') because its pitch is affected by the sharp sign at the beginning of the staff, as in the ***kern*$^*$ encoding pitch must be explicitly represented for every note.

There are other components of notated music that trained models have to learn. For example, the duration of a note or rest is encoded as a token within a note or rest symbol, in both the agnostic and the semantic encoding. A beam is also encoded as a token in a note agnostic symbol. They are not explicitly translated to any semantic symbol, but rather define the durations of beamed notes. Since this establishes a one-to-one correspondence between symbols in both encodings, their translation should be an easy task for the proposed models, which have not been further investigated in this work.

#### 3.2.2. Ambiguous Translation Cases

The translation of the key signature is a challenge in itself for machine learning models, as they have to discriminate when a sharp or a flat is part of a key signature and when it is not. The models must learn that accidentals that are part of a key signature must be all of the same type (sharp or flats), and must be written in a specific order. In addition, they must learn which notes on the staff are affected by the key signature. When a note altered by an accidental appears next to the key signature, it can lead to ambiguous cases where the models (and even humans, in some cases), cannot tell whether the accidental preceding the note is part of the key signature or not. In general, a musician would use information from a broader context than the current staff to tell whether the accidental is part of the key (for example, the key signature found in previous staves). However, in this experimental setup, input instances consist of isolated staves, without song context, which makes it more difficult for the models to correctly translate what is related to a key signature in those cases.

Figure 7 presents such situations. Figure 7a,b shows examples of ambiguous situations where even a human, without a larger context, would struggle to decide whether the accidental next to the note is part of the key signature or not. The other three cases exemplify situations where the models must learn to apply the key signature definition rules mentioned in the previous paragraph, and consider the accidental next to the note as not part of the key signature. These ambiguous cases are seldom found in our corpora. They are present in 0.92% of samples in FMT and 5% of Zaragoza samples. The Camera-PrIMuScorpus does not contain any of these cases, as its samples are synthetic and have been built with a meter sign between the key signature and the beginning of the first measure. Despite having so few cases in our corpora, it is interesting to know how trained models deal with these rare events.

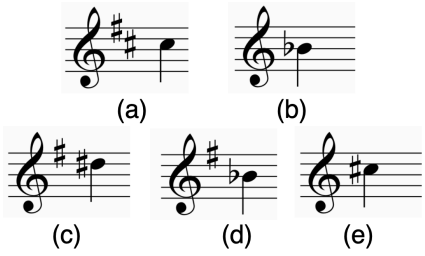



**Figure 7.** (**a**,**b**) Examples of ambiguous key signatures. (**c**–**e**) Cases where key signature definition rules must be applied to consider the accidental next to the note as not part of the key signature.

The presence of staves with absent clef and key in the FMT-C dataset, as discussed in Section 3.1, could also be considered as ambiguous cases that models will have to struggle with, where only 25.9% of the staves in that dataset contain clef and key signature symbols.

### *3.3. Configuration of Machine Translation Systems*

In this part, we introduce and specify, for the sake of reproducibility, the implementation of all the approaches used to perform the automatic translation between the proposed music encodings.

#### 3.3.1. Statistical Approach

For the statistical approach, the well-known Moses toolkit [34] was used. Minimum Error Rate Training (MERT) [35] was the algorithm for tuning the models in the log-linear combination.

The optimal configuration for the task was determined by running a grid search on a series of parameters in the FMT corpus, as this is a medium-sized dataset that offers a good balance between the amount of data and the computational cost of running an exhaustive evaluation. As a result of this evaluation, the order of the language model was set to 7 and the maximum size of the phrases in the translation model was set to 5.

#### 3.3.2. Sequence to Sequence with Attention

In the sequence to sequence implementation, hereby referred to as the Seq2Seq-Attn approach, we implemented a basic encoder/decoder recurrent architecture (see Figure 3 for graphical details) with long short-term memory layers. The optimal configuration of the network depends on the word embedding size and the depth of the recurrent layers, where we try to generate narrow vectors, always smaller than the size of the vocabularies used, which group the necessary information to establish relationships between the different tokens that compose the network entries. After performing a selection process on the best subsets to determine these parameters, we found that the network obtains, in general, its best performance with a word embedding of 64 and a depth of the recurrent layers of 128. A description of this implementation is shown in Figure 8.

| Input ($l_0 \times \Sigma_a$) |
|---|
| Embedding(64) |
| LSTM(128) |
| Input($l_1 \times \Sigma_s$) |
| LSTM(128) |
| GlobalAttention() |
| Dense($\Sigma_s$) |
| Softmax() |

**Figure 8.** Implementation scheme of the Seq2Seq-Attn approach, where $l_n$ represents the sample length, $\Sigma_a$ the size of the agnostic vocabulary and $\Sigma_s$ is the size of the semantic vocabulary.

#### 3.3.3. The Transformer

The transformer implementation was based on the architecture defined in the original work and depicted in Figure 4. However, in this work we produced a simpler model in number of layers and embedding sizes, as the reference implementation was designed to handle languages with bigger and more complex vocabularies. As we did with the Seq2Seq-Attn approach, we found the optimal configuration parameters for the network with the best subset selection process. The layer implementation of this model is described in Figure 9.

| Input ($l_0 \times \Sigma_a$) |
|---|
| Embedding(64) |
| PositionalEncoding($l$) |
| MHA(64,8) |
| LayerNormalization() |
| Dense(128) |
| ReLU() |
| Dense(64) |
| LayerNormalization() |
| Input ($l_1 \times \Sigma_s$) |
| Embedding(64) |
| PositionalEncoding($l$) |
| MHA(64,8) |
| LayerNormalization() |
| MHA(64,8) |
| LayerNormalization() |
| Dense(128) |
| ReLU() |
| Dense(64) |
| LayerNormalization() |
| Dense($\Sigma_s$) |
| Softmax() |

**Figure 9.** Implementation scheme of the transformer model, where MHA represents the multi-headed attention layers, which take as a first parameter their size and as a second parameter the number of heads that the attention matrix has to be split in. $l$ represents the sample length; $\Sigma_a$ is the size of the agnostic vocabulary, even of not being a one-hot encoded input; and $\Sigma_s$ is the size of the semantic vocabulary.

## 4. Results

The first part of this section explains the metrics used for performance evaluation. Next, the results of the experiments are presented, and finally a discussion of the error in the translation categories presented above is given.

### *4.1. Evaluation Process and Metrics*

Experiments have been run to evaluate the accuracy in translating from the agnostic notation into ***kern** for the three approaches described in Section 2: the statistical approach (see Section 2.2), the neural approach based on the sequence to sequence architecture with attention (see Section 2.3.1) and the neural approach based on the Transformer architecture (see Section 2.3.2).

The translation performance was evaluated by computing the symbol error rate (SER) between each translation hypothesis $h$ and the corresponding reference translation $r$, which is defined as: $d_e(h,r)/|r|$, where $d_e(h,r)$ is the symbol-level Levenshtein's edit distance between $r$ and $h$, and $|r|$ is the number of symbols in $r$.

Every dataset is a compilation of images, each containing one or two pages. Each page was segmented into regions of types such as title, staff and lyrics, of which only the agnostic encodings of staves were considered. In order to obtain a robust approximation error, 10-fold partitions were constructed for each dataset. For each set, the list of all pages in it were shuffled and sequentially assigned to one fold, obtaining size-balanced folds with randomly assigned pages. Then, for each page, all different encodings were created and written in a JSON file so that all experiments used the same sample distribution. Each approach was evaluated by iteratively using eight parts as the training set; one part was used as the validation set; and the remaining part was the test set.

### *4.2. Experimental Results*

The results presented in the Table 3 strongly depend on the corpus used for evaluation. As can be seen, for all the three approaches evaluated, the best results were obtained for the Camera-PrIMuS corpus, for which the transformer outperformed the Seq2Seq-Attn architecture by 1.5% SER, and the SMT approach by more than 1% SER. On the other end of performance, the worst results were obtained for the Zaragoza corpus, with which the SMT approach outperformed the transformer and Seq2Seq-Attn architectures by 51% SER and 56% SER, respectively. In the case of FMT, the Seq2Seq-Attn and the SMT approaches obtained almost the same result, and outperformed the transformer by about 5% SER.

**Table 3.** Results obtained in terms of average SER and standard deviation on the 10-fold cross-validation samples for the three corpora used for evaluation. Results for the three approaches evaluated on the task of translating standard agnostic input into ***kern*$^*$ are reported. Additional results are presented in the last row using SMT to translate contextual agnostic input into ***kern*$^*$.

| | **Camera-PrIMuS** | **FMT** | **Zaragoza** |
|---|---|---|---|
| SMT | 23.7 ± 0.3% | **9.6 ± 3.8%** | 69 ± 12% |
| Seq2Seq-Attn | 2.04 ± 0.09% | 9.8 ± 3.2% | 89.5 ± 1.6% |
| The Transformer | **0.50 ± 0.06%** | 15.3 ± 3.9% | 86.3 ± 5.1% |
| SMT Contextual | 1.58 ± 0.05% | 11.1 ± 4.3% | **35 ± 16%** |

Regarding the SMT method, it should be noted that the results obtained differ noticeably depending on the type of agnostic notation used in the different corpora. For the Camera-PrIMuS and the Zaragoza corpora, the SER increased dramatically when contextual information was only provided at the beginning of the segment (standard agnostic notation). This result was expected, and confirms the hypothesis that when context is relevant to translating a symbol that appears far from the contextual information (i.e., the clef and key signature), this approach is unable to translate it properly. However, this limitation is overcome in the neural approaches through the attention mechanism. The only exceptions in this regard are the results obtained for the FMT corpus, which showed very similar results for both the contextual and the non-contextual agnostic notations when applying the statistical approach. The explanation for these contradictory results lies in two key aspects related to the addition of contextual information to the agnostic notation: the number of unknown symbols in the test set (see Table 1) and the improvement of the performance for specific types of errors related to contextual information (see Table 4).

**Table 4.** Masked SER for some translation categories. The overall SER is also shown for easy comparison.

| | **Contextual SMT** | **SMT** | **Seq2Seq-Attn** | **The Transformer** |
|---|---|---|---|---|
| Camera-PrIMuS | | | | |
| Pitch | 1.28% | 22.74% | 1.70% | 0.42% |
| Dotted notes | 0.90% | 11.73% | 2.60% | 0.33% |
| Pitch altered by key | 0.97% | 16.31% | 1.73% | 0.27% |
| Overall | 1.58% | 23.70% | 2.04% | 0.50% |
| FMT | | | | |
| Pitch | 10.43% | 9.03% | 11.34% | 16.32% |
| Dotted notes | 14.22% | 11.21% | 19.18% | 18.32% |
| Pitch altered by key | 15.91% | 20.63% | 21.10% | 18.74% |
| Slurs and ties | 14.11% | 11.35% | 15.03% | 19.33% |
| Overall | 11.10% | 9.61% | 9.75% | 15.30% |
| Zaragoza | | | | |
| Pitch | 23.67% | 68.43% | 91.78% | 90.63% |
| Dotted notes | 24.32% | 68.47% | 92.43% | 91.35% |
| Pitch altered by key | 24.76% | 63.59% | 94.17% | 91.74% |
| Slurs and ties | 50.00% | 80.00% | 90.00% | 90.00% |
| Overall | 35.31% | 68.60% | 89.50% | 86.31% |

As discussed in Section 4.3 below, the improvement in the translation accuracy for FMT when using contextual agnostic notation was more modest than for Zaragoza and Camera-PrIMuS. However, the ratio of unknown words in the FMT test set increased from 0.13 to 0.28. As a result, FMT is the only case in which using contextual agnostic notation not only did not improve the final result, but even slightly worsened it.

Regarding the two neural approaches evaluated, the transformer clearly obtained the best results on the Camera-PrIMuS corpus. However, none of the neural approaches seemed to outperform the SMT approach for the FMT and Zaragoza corpora. This may be explained by the fact that these two corpora are substantially smaller than Camera-PrIMuS, and neural approaches do not seem to be able to learn enough from small amounts of training data.

To test this hypothesis, we ran an additional experiment by taking subsamples of 1000 to 5000 segment pairs from the Camera-PrIMuS corpus and used them to evaluate the SMT approach (the best performance on small corpora) and the transformer approach (the best performance on Camera-PrIMuS). As can be seen in Figure 10, the SMT approach slightly outperformed the transformer on the smaller evaluation subsample. However, when the size of the dataset increased, the transformer became more accurate than SMT. In fact, as the size of the dataset got larger, the difference between both approaches became larger and the deviations got smaller, so the differences are more significant. These results clearly confirm our hypothesis that with enough data, the transformer clearly outperforms the SMT approach.

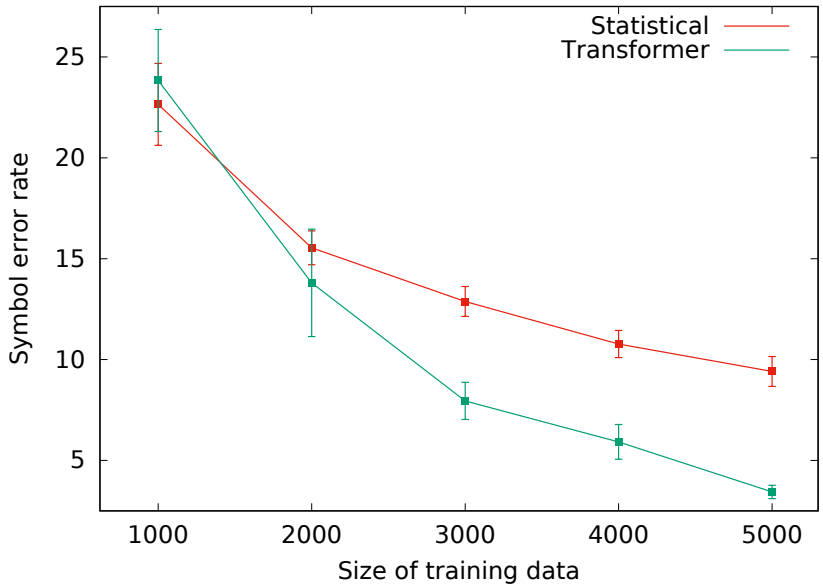



**Figure 10.** Mean symbol error rate (SER) and standard deviation, represented by vertical bars, obtained when training the statistical and the transformer neural approaches with subsets of the PRIMUS corpus of different sizes.

Other relevant dimensions to take into account in the comparison of these machine translation approaches are the amount and type of computational resources required by each of them. Table 5 shows the total time consumed by each approach, the peak memory and the amount of CPU/GPU used to train and to translate one fold of each of the evaluated datasets. Note that neural approaches require a GPU to be trained, while statistical models are able to be trained on CPUs. For this reason, SMT models were trained and run on a machine with an AMD Opteron(tm) 6128 CPU with 8 cores and 64GB of RAM, and the neural models were trained on a machine with an Intel® Xeon® E3-1230 v5 CPU with 8 cores, 32GB of RAM and a GeForce GTX 1080 GPU. It is worth mentioning that the time performance of the neural models is highly dependent on the GPU model used; for example, we tested the Seq2Seq-Attn model with a NVIDIA RTX 2080 GPU, and reduced the training time by 25%, thereby only taking one day to train a model.

**Table 5.** Total time and peak memory consumed by each the three approaches evaluated (SMT, Seq2Seq-Attn, and the Transformer) to train and to translate one fold of each of the three datasets used for evaluation (Camera-PrIMuS, FMT, and Zaragoza).

| | | Training | | Translating | |
|---|---|---|---|---|---|
| **Approach** | **Data Set** | **Time** | **Memory** | **Time** | **Memory** |
| SMT | Camera-PrIMuS | 12 h | 31 MB | 2 h | 40 MB |
| | FMT | 6′ | 31 MB | 13″ | 38 MB |
| | Zaragoza | 6′ | 30 MB | 10″ | 37 MB |
| Seq2Seq-Attn | Camera-PrIMuS | 4 d | 317 MB | 10″ | 317 MB |
| | FMT | 3 h | 317 MB | 9″ | 317 MB |
| | Zaragoza | 5′ | 317 MB | 7″ | 317 MB |
| The Transformer | Camera-PrIMuS | 5 h | 385 MB | 7″ | 385 MB |
| | FMT | 13′ | 385 MB | 5″ | 385 MB |
| | Zaragoza | 5′ | 385 MB | 3″ | 385 MB |

### *4.3. Error Analysis of Translation Categories*

In order to gain some insights on the performance of the proposed models in some of the translation categories discussed in Section 3.2.1, the positions of notes affected by the associated phenomena were identified and annotated for each of the three corpora,

based on ground truth data. Then, the SER was computed taking into account only these notes for each of the translation categories analyzed. For that purpose, we computed a *masked SER*, for which only the errors in the target notes were taken into account. For the masked version of SER, we aligned the translation hypothesis $h$ and the ground-truth reference $r$ as usual, but took into account only the symbols involved in the translation category studied to compute the error rate.

Table 4 presents the masked SER for the translation categories of interest. Note, however, that due to how this metric is computed at the symbol level, results in each category were most probably influenced by errors due to phenomena from other categories. For example, an error predicting the pitch of a note would also compute as an error in the dotted notes, and the pitch altered by key categories. Hence, these results have to be taken as an inkling of which categories may contribute more to the overall SER, and deserve further study.

For the Camera-PrIMuS corpus, it is evident that the availability of contextual information about the current clef and key signature of each note drastically improved the performance of the statistical (SMT) method, approaching the performance figures of neural models. The improvement for the SMT has occurred despite the increase of the vocabulary size of the corpus more than sixteen times. Pitch and pitch altered by key categories were expected to improve when introducing the tonal context in the input symbols. The fact that there was also a great improvement in the dotted notes category, which a priori should not be affected by contextual information, as it is irrelevant for computing the duration of notes, was most probably due to the fact that errors in this category were indeed due to errors related to pitch prediction.

The results for the FMT corpus revealed that the pitch altered by key issue is the hardest to learn for all models except the Transformer. Note that the use of contextual information for the SMT approach does not provide a clear improvement of the performance. In fact, contextual SMT seems to perform worse than its non-contextual counterpart for some translation categories. This could be due to two unrelated factors that, combined, could explain that lack of improvement: first, remember from Table 1 that the size of the vocabulary with contextual information almost doubles the size of the non-contextual one, while the size of the corpus is relatively small. Therefore, the probability of the model to find a symbol never seen during training is significant. In fact, a manual investigation of the translation errors revealed that a significant amount of them where due to the presence of such unknown symbols, which the model was unable to translate. Note that SMT implements smoothing strategies to deal with words in the source language that have never been seen during training. In these cases, the unseen word remains untranslated. Moreover, the presence of contextual information in agnostic note symbols was devised to help improve the prediction of pitch (which depends on the clef) and pitch altered by key (which obviously depends on the key signature). However, the only clef present in this corpus is the G-clef, so the models are not able to establish any relationship between the clef and the note pitches. They de-facto assume a given vertical position of a symbol on the staff is always translated to the same pitch. In fact, the only actual improvement for the contextual SMT model is found in the pitch altered by key category.

The second factor that explains the lack of overall improvement for the contextual SMT model on FMT is the fact that around one third of the pieces in this corpus have staves without clef and key signature, as discussed in Section 3.1, and detailed in Table 2. Most samples from these pieces do not have clef and key signature context, which does not help to improve the performance of the statistical model. The neural models also struggled in translating this corpus, at least when compared with their performance figures on Camera-PrIMuS, and this was most probably due to the small size of the corpus. In particular, the Seq2Seq-Attn model performance on the pitch altered by key category for the FMT corpus has been investigated. The model can successfully predict most altered-by-key pitches, even with of absent clef and key signatures. However, this is mostly the case

when there is only one accidental in the key signature. The presence of more complex key signatures in the samples tend to raise the error rate in this category.

The Zaragoza corpus was the smallest corpus in this study, and all the models performed poorly on it. However, a significant improvement in performance was obtained for the contextual SMT over SMT. In this case, every staff in the corpus contained a clef and a key signature, which definitively helped SMT to perform better, but still with a high error rate. The standard deviation was especially high for the results in this dataset, which points out that the models are far from robust: when translating folds that were slightly different to training data (for example, as regards the length of the staffs or the amount of unseen symbols during training), the translation failed and produced results comparable to those obtained by the neural approaches. As for the poor performance of the neural models, it is clear that they do not have enough instances to tune their parameters, as revealed by a close look at their outputs, where it is clear that they learned almost nothing about the syntax and semantics of the input.

In general, errors due to the insertion and deletion of symbols in the predicted output revealed that the models were not only able to learn the input language, but also to build a language model from the data which, in the case of music, means learning the structure of phrases, or how melodies are composed. In some cases, the models inserted a note that was not present in the input, or deleted another, in what appear to have been attempts to correct poorly formed sentences (melodies). This could be a potential advantage when dealing with agnostic representations obtained directly from the object recognition modules, as errors in the recognition phase could be present in the input. The machine learning models of the encoding phase, adequately trained, could introduce an opportunity to correct those errors.

## 5. Conclusions

In this paper, we have studied the application of machine translation (MT) techniques to performing the encoding step in an optical music recognition pipeline, which had not been properly addressed prior to our study. This step consists of converting a graphics-based sequence, called agnostic encoding, into a standard digital music notation encoding, called semantic encoding.

We addressed this task by presenting three solutions, one based on statistical machine translation (SMT) techniques, and two based on neural network systems, namely, a sequence to sequence model with attention mechanisms and the transformer model, which is currently the state-of-the-art neural approach in natural language processing and machine translation tasks.

From the experimental results, we observed that the transformer and SMT approaches performed the best, depending on the size of the corpus. From this first analysis, we observed that the model that currently produced the best language model for the translation task was the transformer, when provided with enough data. However, in cases where the corpus lacked sufficient information for this model to converge, the SMT approach obtained interesting results that made it a valid solution for these specific cases. However, this result needs to be put in context. From a practical point of view, it should also be noted that the SMT solution requires feature engineering processes to obtain its full performance. The design of the contextual agnostic notation can be considered an example of this, which could be a drawback in a practical scenario, as the annotation of the corpus can be difficult to perform.

We consider then that, although they are not the best solutions to the problem, neural approaches are practical, because they do not require this feature engineering process. They can also be integrated in a full OMR pipeline, as they also share technological properties with the rest of the steps that take part of the recognition processes. In other words, the SMT approach is the best solution if the corpus lacks sufficient data, but it requires additional process of the dataset in order to achieve good results. The technical decision on which model should be used depends entirely on the size of the corpus

and the assumption of additional processing costs to train a SMT model when there is insufficient data.

Apart from the comparisons between solutions, we get a clear idea from this work: the application of machine translation to solve this aspect of the optical music recognition process is currently a feasible solution, as it provides robust and competitive results on different corpora when there is enough data.

This article establishes a baseline for future work on this topic, as the coding step in OMR deserves more attention than is found in the literature. This research reveals new problems, questions and challenges that have to be further addressed in the future. The most relevant one is the application of automatic translation techniques to a complete OMR pipeline, where the input is the result of automatic image recognition and may not be accurate. Research in this direction would definitely solve the question of how these approaches improve the state-of-the art complete pipelines and how beneficial may be in specific situations, such as data sparsity or unbalanced corpora.

As part of this research, the study of the consistency of the proposed machine translation models should be further evaluated when there are recognition errors in the input. One approach worth exploring is to propose an input data augmentation algorithm to improve the robustness of the models in presence of inconsistencies.

We think it would also be interesting to deepen into the musical semantics that the models are learning during the training process, in order to produce better language models that can improve the results reported in this article.

As for the different translation technologies evaluated in this work, the transformer seems to be one of the most promising in the field of natural language processing. There is likely to be room for improvement in the field of translation between music notations. One obvious path in this direction is to feed a transformer with more than one line at the output; this strategy, which is frequently used in the translation between natural languages, like the *mBART* model [36], one of the most promising state-of-the-art methods currently used for machine translation, which is able to take several sentences at a time as an input, up to a maximum length of 512 tokens, could help to deal with scores in which the clefs and the keys are not explicitly provided in every staff, and are assumed to propagate from staff to staff.

**Author Contributions:** Conceptualization, A.R.-V., M.E.-G. and D.R.; methodology, A.R.-V. and M.E.-G.; software, A.R.-V. and M.E.-G.; validation, A.R.-V., M.E.-G., D.R. and P.J.P.d.L.; formal analysis, M.E.-G.; investigation, A.R.-V., M.E.-G. and D.R.; resources, D.R. and P.J.P.d.L.; data curation, D.R. and P.J.P.d.L.; writing, A.R.-V, M.E.-G., D.R., P.J.P.d.L. and J.M.I; visualization, M.E.-G. and D.R.; supervision, D.R. and J.M.I.; funding acquisition, J.M.I. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Spanish Ministry HISPAMUS project TIN2017-86576-R, partially funded by the EU, and by the Generalitat Valenciana through project GV/2020/030.

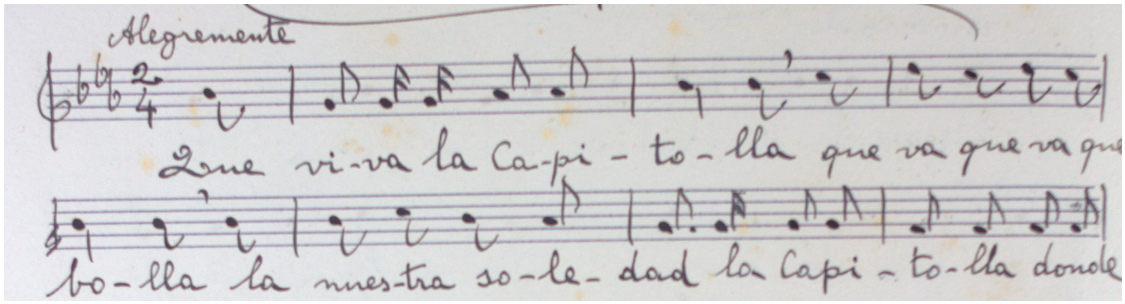
**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data utilized in this work are available to the interested researchers upon request to the authors.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A




(**a**) La Capitolla

```
clef.G:L2, accidental.flat:L3, accidental.flat:S4, accidental.flat:S2, digit.2:L4,
digit.4:L2, note.eighth_down:L3, verticalLine:L1, note.eighth_up:L2, note.sixteenth_up:L2,
note.sixteenth_up:L2, note.eighth_up:S2, note.eighth_up:S2, verticalLine:L1, note.quarter_down:L3,
note.eighth_down:L3, note.eighth_down:S3, verticalLine:L1, note.eighth_down:L4, note.eighth_down:S3,
note.eighth_down:L4, note.eighth_down:S3
```

(**b**) Agnostic encoding of the top staff

```
clef.G:L2, accidental.flat:L3, accidental.flat:S4, accidental.flat:S2, digit.2:L4,
digit.4:L2, note.eighth_down@clef.G:L2@3b:L3, verticalLine:L1, note.eighth_up@clef.G:L2@3b:L2,
note.sixteenth_up@clef.G:L2@3b:L2, note.sixteenth_up@clef.G:L2@3b:L2, note.eighth_up@clef.G:L2@3b:S2,
note.eighth_up@clef.G:L2@3b:S2, verticalLine:L1, note.quarter_down@clef.G:L2@3b:L3,
note.eighth_down@clef.G:L2@3b:L3, note.eighth_down@clef.G:L2@3b:S3, verticalLine:L1,
note.eighth_down@clef.G:L2@3b:L4, note.eighth_down@clef.G:L2@3b:S3, note.eighth_down@clef.G:L2@3b:L4,
note.eighth_down@clef.G:L2@3b:S3
```

(**c**) Agnostic encoding of the top staff including context

```
note.quarter_down:L3, note.eighth_down:L3, note.eighth_down:L3, verticalLine:L1,
note.eighth_down:L3, note.eighth_down:S3, note.eighth_down:L3, note.eighth_up:S2, verticalLine:L1,
note.eighth_up:L2, dot:S2, note.sixteenth_up:L2, note.eighth_up:L2, note.eighth_up:L2,
verticalLine:L1, note.eighth_up:S1, note.eighth_up:S1, note.eighth_up:S1, note.eighth_up:S1
```

(**d**) Agnostic encoding of the bottom staff

```
note.quarter_down:L3, note.eighth_down:L3, note.eighth_down:L3, verticalLine:L1,
note.eighth_down:L3, note.eighth_down:S3, note.eighth_down:L3, note.eighth_up:S2,
verticalLine:L1, note.eighth_up:L2, dot:S2, note.sixteenth_up:L2, note.eighth_up:L2,
note.eighth_up:L2, verticalLine:L1, note.eighth_up:S1, note.eighth_up:S1,
note.eighth_up:S1, note.eighth_up:S1
```

(**e**) Agnostic encoding of the bottom staff including context

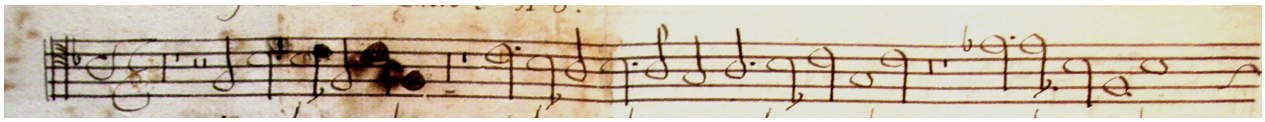
```
**skern *clefG2 *k[b-e-a-] *M2/4 8b-\=
8g/ 16g/ 16g/ 8a-/ 8a-/ =
4b-\8b-\8cc\=
8dd\8cc\8dd\8cc\
```

(**f**) Semantic encoding of the top staff (in order to save space the space separator has been used instead of the actual end of line)

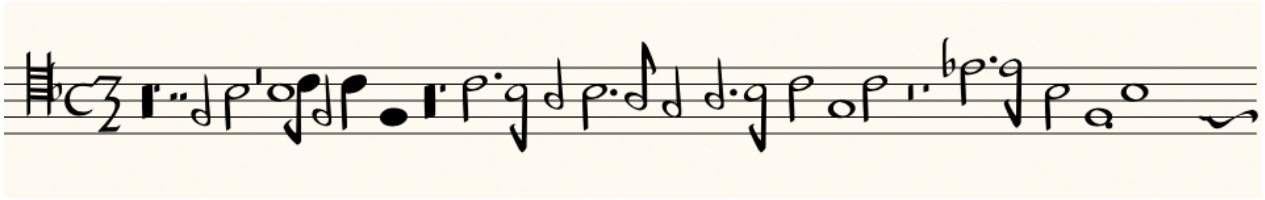
```
**skern = 4b-\8b-\8b-\=
8b-\8cc\8b-\8a-/ =
8.g/ 16g/ 8g/ 8g/ = 8f/ 8f/ 8f/ 8f/
```

(**g**) Semantic encoding of the bottom staff (in order to save space the space separator has been used instead of the actual end of line)

**Figure A1.** Sample from the FMT corpus (**a**) [37], where we observe both the raw and contextual agnostic encoding of the both staves (**b**–**e**) and their corresponding ***kern** sequence (**f**,**g**).



(**a**) Notation staff



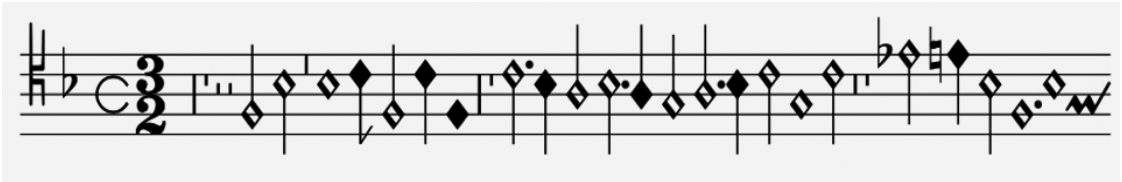
(**b**) Agnostic visual representation

```
clef.C:L4, accidental.flat:S3, metersign.CZ:L3, rest.longa2:L2, rest.whole:L4, rest.half:L3, rest.half:L3,
note.half_up:L2, note.half_down:S3, rest.breve:L4, note.whole:S3, note.eighth_down:L4, note.half_up:L2,
note.quarter_down:L4,note.wholeBlack:L2, rest.longa2:L2, rest.whole:L4, note.half_down:L4, dot:S4,
note.eighthVoid_down:S3, note.half_up:L3, note.half_down:S3, dot:S3, note.eighthVoid_up:L3,
note.half_up:S2, note.half_up:L3, dot:S3, note.eighthVoid_down:S3, note.half_down:L4, note.whole:S2,
note.half_down:L4, rest.breve:L3, rest.whole:L4, accidental.flat:L5, note.half_down:L5, dot:S5,
note.eighthVoid_down:L5, note.half_down:S3, note.whole:L2, dot:S1, note.whole:S3, custos:L2
```

(**c**) Agnostic encoding as output by the symbol recognition phase of a OMR system

```
clef.C:L4, accidental.flat:S3, metersign.CZ:L3, rest.breve:L2, rest.whole:L3, rest.half:L2,
note.half_up@clef.C:L4@1b:L2, note.half_up@clef.C:L4@1b:L2, verticalLine:L1,
note.half_up@clef.C:L4@1b:S2, dot:S2, note.eighthVoid_down@clef.C:L4@1b:L3,
note.half_down@clef.C:L4@1b:S3, dot:S3, note.eighth_down@clef.C:L4@1b:S3,
note.wholeBlack@clef.C:L4@1b:L3, note.half_up@clef.C:L4@1b:S2, dot:S2,
note.eighthVoid_up@clef.C:L4@1b:S2, note.half_up@clef.C:L4@1b:L2, dot:S2,
note.eighth_up@clef.C:L4@1b:L2, note.wholeBlack@clef.C:L4@1b:S0, rest.whole:L3, rest.half:L2,
note.half_up@clef.C:L4@1b:S2, note.half_up@clef.C:L4@1b:S2, note.half_up@clef.C:L4@1b:L3, dot:S3,
note.eighthVoid_down@clef.C:L4@1b:S3, note.half_down@clef.C:L4@1b:L4, dot:S4,
note.eighthVoid_down@clef.C:L4@1b:L4, dot:S4, accidental.sharp:L1, note.wholeBlack@clef.C:L4@1b:S3,
note.half_down@clef.C:L4@1b:L3, rest.half:L4, rest.half:L4, verticalLine:L1, rest.half:L3,
note.half_down@clef.C:L4@1b:S4, note.half_up@clef.C:L4@1b:L1, note.whole@clef.C:L4@1b:S2,
note.half_up@clef.C:L4@1b:S2, custos:L4
```

(**d**) Agnostic encoding including context



(**e**) Staff semantic visual representation

```
*clefC4 *k[b-] *met(C32) Lr_2 sr_4 Mr_3 Mr_3 m~F m~B- Sr_4 sB- Uc m~F M~c s~F Lr_2 sr_4 m.~c mB-
m~A m.~B- mA m~G m.~A mB- m~c sG m~c Sr_3 sr_4 m.~e- me m~B- s.F sB- *custosF
```

(**f**) Staff semantic encoding (in order to save space the space separator has been used instead of the actual end of line)

**Figure A2.** (**a**) Sample of a music staff from the Zaragoza corpus. (**b**) This staff is a rendering of the music symbols found in the image by a OMR tool, as encoded in (**c**,**d**). (**e**,**f**) The semantic encoding of the sample.

(**a**) RISM ID no. 225002139, Incipit 1.1.2. *Wiegenlied*, Franz von Holstein

```
clef.G:L2, accidental.sharp:L5, accidental.sharp:S3, accidental.sharp:S5,
accidental.sharp:L4, digit.6:L4digit.8:L2, multirest:L3, digit.4:S5, verticalLine:L1,
note.beamedRight1_up:L2, dot:S2, note.beamedLeft2_up:S2, note.eighth_up:L2, note.quarter_up:S1,
note.sixteenth_up:S1, note.sixteenth_up:L2, verticalLine:L1, note.beamedRight1_up:S2,
note.beamedLeft1_down:S3, note.eighth_down:L3, note.quarter_up:L2, dot:S2, verticalLine:L1
```

(**b**) Agnostic encoding

```
clef.G:L2, accidental.sharp:L5, accidental.sharp:S3, accidental.sharp:S5,
accidental.sharp:L4, digit.6:L4digit.8:L2, multirest:L3, digit.4:S5, verticalLine:L1,
note.beamedRight1_up@clef.G:L2@4#:L2, dot:S2, note.beamedLeft2_up@clef.G:L2@4#:S2,
note.eighth_up@clef.G:L2@4#:L2, note.quarter_up@clef.G:L2@4#:S1, note.sixteenth_up@clef.G:L2@4#:S1,
note.sixteenth_up@clef.G:L2@4#:L2, verticalLine:L1, note.beamedRight1_up@clef.G:L2@4#:S2,
note.beamedLeft1_down@clef.G:L2@4#:S3, note.eighth_down@clef.G:L2@4#:L3,
note.quarter_up@clef.G:L2@4#:L2, dot:S2, verticalLine:L1
```

(**c**) Agnostic encoding including context

```
**skern *clefG2 *k[f#c#g#d#] *M6/8 rr4 =
8.g# 16a 8g# 4f# 16f# 16g# =
8a 8cc# 8b 4.g# =
```

(**d**) Semantic encoding (in order to save space the space separator has been used instead of the actual end of line)

**Figure A3.** Sample from an incipit of the Camera-PrIMuS corpus (**a**), where we can observe the raw agnostic encoding of the shown staff (**b**), the contextual agnostic sequence (**c**) produced for the SMT model and its semantic encoding representation in ***kern** (**d**).

## References


1. Calvo-Zaragoza, J.; Hajič J., Jr.; Pacha, A. Understanding Optical Music Recognition. *ACM Comput. Surv.* **2020**, *53*. [CrossRef]
2. Rebelo, A.; Cardoso, J. Staff Line Detection and Removal in the Grayscale Domain. In Proceedings of the 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013; pp. 57–61. [CrossRef]
3. Gomez, A.A.; Sujatha, C.N. Optical Music Recognition: Staffline Detection and Removal. *Int. J. Appl. Innov. Eng. Manag.* **2017**, *6*, 48–58.
4. Wen, C.; Rebelo, A.; Zhang, J.; Cardoso, J. Classification of optical music symbols based on combined neural network. In Proceedings of the International Conference on Mechatronics and Control, Jinzhou, China, 3–5 July 2014; pp. 419–423.
5. Pacha, A.; Eidenberger, H. Towards a Universal Music Symbol Classifier. In Proceedings of the 14th International Conference on Document Analysis and Recognition, Kyoto, Japan, 9–15 November 2017; IAPR TC10 (Technical Committee on Graphics Recognition); IEEE Computer Society: Washington, DC, USA, 2017; pp. 35–36.
6. Rossant, F.; Bloch, I. Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection. *EURASIP J. Adv. Signal Process.* **2006**, *2007*, 081541. [CrossRef]
7. Liu, X.; Zhou, M.; Xu, P. A Robust Method for Musical Note Recognition. In Proceedings of the 14th International Conference on Computer-Aided Design and Computer Graphics, Xi'an, China, 26–28 August 2015; pp. 212–213.
8. Calvo-Zaragoza, J.; Toselli, A.H.; Vidal, E. Handwritten Music Recognition for Mensural notation with convolutional recurrent neural networks. *Pattern Recognit. Lett.* **2019**, *128*, 115–121. [CrossRef]
9. Cuthbert, M.S.; Ariza, C. Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data. In Proceedings of the Proceedings of International Society for Music Information Retrieval Conference, Utrecht, The Netherlands, 9–13 August 2010; pp. 637–642.
10. Sapp, C.S. Verovio Humdrum Viewer. In Proceedings of the Proceedings of Music Encoding Conference (MEC), Tours, France, 16–19 May 2017.
11. Couasnon, B. DMOS: A generic document recognition method, application to an automatic generator of musical scores, mathematical formulae and table structures recognition systems. In Proceedings of the Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA, 10–13 September 2001; pp. 215–220.

12. Thomae, M.E.; Ríos Vila, A.; Calvo-Zaragoza, J.; Rizo, D.; Iñesta, J.M. Retrieving Music Semantics from Optical Music Recognition by Machine Translation. In Proceedings of the Retrieving Music Semantics from Optical Music Recognition by Machine Translation, Medford, MA, USA, 26–29 May 2020.
13. Roland, P. The Music Encoding Initiative (MEI). Available online: http://xml.coverpages.org/MAX2002-PRoland.pdf (accessed on 1 February 2021).
14. Hankinson, A.; Roland, P.; Fujinaga, I. The Music Encoding Initiative as a Document-Encoding Framework. In Proceedings of the 12th International Society for Music Information Retrieval Conference, Miami, FL, USA, 24–28 October 2011.
15. Good, M.; Actor, G. Using MusicXML for File Interchange. In Proceedings of the Web Delivering of Music, International Conference on, Leeds, UK, 15–17 September 2003; p. 153.
16. Huron, D. Humdrum and Kern: Selective Feature Encoding. In *Beyond MIDI: The Handbook of Musical Codes*; MIT Press: Cambridge, MA, USA, 1997; pp. 375–401.
17. Calvo-Zaragoza, J.; Rizo, D. Camera-PrIMuS: Neural End-to-End Optical Music Recognition on Realistic Monophonic Scores. In Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, 23–27 September 2018, pp. 248–255.
18. Pacha, A.; Calvo-Zaragoza, J.; Hajič, J., Jr. Learning Notation Graph Construction for Full-Pipeline Optical Music Recognition. In Proceedings of the 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 4–8 November 2019; pp. 75–82.
19. Calvo-Zaragoza, J.; Rizo, D. End-to-End Neural Optical Music Recognition of Monophonic Scores. *Appl. Sci.* **2018**, *8*, 606–623. [CrossRef]
20. Ríos-Vila, A.; Calvo-Zaragoza, J.; Rizo, D. Evaluating Simultaneous Recognition and Encoding for Optical Music Recognition. In Proceedings of the 7th International Conference on Digital Libraries for Musicology, Montréal, QC, Canada, 16 October 2020; DLfM 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 10–17.
21. Koehn, P. *Statistical Machine Translation*; Cambridge University Press: Cambridge, UK, 2009.
22. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *arXiv* **2014**, arXiv:1409.3215.
23. Luong, M.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. *arXiv* **2015**, arXiv:1508.04025.
24. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165.
25. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
26. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
27. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
28. Rizo, D.; Pascual-León, N.; Sapp, C. White Mensural Manual Encoding: From Humdrum to MEI. *Cuad. Investig. Music.* **2018**, 373–393. [CrossRef]
29. RISM Code "E-Zac". Available online: https://rism.info/ (accessed on 1 February 2021).
30. Calvo-Zaragoza, J.; Rizo, D.; Iñesta, J.M. Two (Note) Heads Are Better Than One - Pen-Based Multimodal Interaction with Music Scores. In Proceedings of the International Society for Music Information Retrieval Conference, New York, NY, USA, 7–11 August 2016; pp. 509–514.
31. Rizo, D.; Calvo-Zaragoza, J.; Iñesta, J. MuRET: A music recognition, encoding, and transcription tool. In Proceedings of the 5th International Conference on Digital Libraries for Musicology (DLfM'18), Paris, France, 28 September 2018; pp. 52–56.
32. Keil, K.; Ward, J.A. Applications of RISM data in digital libraries and digital musicology. *Int. J. Digit. Libr.* **2017**, *50*, 199. [CrossRef]
33. Fondo de Música Tradicional IMF-CSIC. Available online: https://musicatradicional.eu/es/home (accessed on 1 February 2021).
34. Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; et al. Moses: Open Source Toolkit for Statistical Machine Translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, Prague, Czech Republic, 25–27 June 2007; Association for Computational Linguistics: Stroudsburg, PA, USA, 2007; pp. 177–180.
35. Och, F.J. Minimum Error Rate Training in Statistical Machine Translation. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan, 7–12 July 2003; Association for Computational Linguistics: Stroudsburg, PA, USA, 2003; pp. 160–167. [CrossRef]
36. Liu, Y.; Gu, J.; Goyal, N.; Li, X.; Edunov, S.; Ghazvininejad, M.; Lewis, M.; Zettlemoyer, L. Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 726–742. [CrossRef]
37. Ros-Fábregas, E.; Mazuela-Anguita, A. La Capitolla. Fondo de Música Tradicional IMF-CSIC. Avalible online: https://musicatradicional.eu/es/piece/1103 (accessed on 1 Feruary 2021).