Music generation with Deep Learning: a systematic review

Abstract. Music is one of the most intrinsic arts of human culture. Being able to stir our emotions, the perception of music says a lot about what makes us human. Our goal is to investigate the current state of artificial music generation through Deep Learning. “What types of deep generative models are mainly used to generate music and how do they work? How can we define the quality of the results? What musical aspects and forms are being generated? What are the ethical and legal implications for the music industry and culture when a machine produces music?” These are the questions we will address in this review.

Keywords: Music Generation · Deep Learning · GAN · LSTM · Transformer · VAE · Deep Generative Model

1 Introduction

Music has been part of human culture for centuries, from Beethoven’s classical compositions to modern-day popular music that is played on radio and is successful at the charts. However, with the advancement of AI, the field of music composition has shifted towards machine-generated music. This has prompted significant debate about the quality, authenticity, and various ways a machine can produce music. To address these concerns, this paper systematically reviews the literature on artificial music generation via deep learning in the past decade, in order to synthesize the state of the art of this technology, find gaps in the studies, and guide future work in this area.

Our methodology involves identifying relevant literature for this review, this process includes establishing eligibility criteria, developing a comprehensive search strategy, gathering data from the sources identified, and combining the information. Through this method, we hope to provide insightful information and increase our understanding of the current state of music generation through machine learning, in terms of: the architectural features of the models; types of music produced; the criteria and metrics for evaluating the results; ways of representing the music; and the debates about authenticity and ethical issues raised in the artistic environment, as well as the impact on society. In short, we intend to analyze whether the current state of music generation by Deep Learning is sufficiently developed and established.

2 Methods

2.1 Eligibility criteria

One selection criteria was to include only scientific articles with a minimum publication year of 2010, since we assumed that this decade was the beginning of phase 3 of Big Data [1], in which Deep Learning had the greatest impact. In phase 3, the availability of data has continued to increase thanks to mobile devices, sensors, and the emergence of the Internet of Things. This way, we can ensure an up-to-date overview. The review used studies in which deep generative models were used as the deep learning model, discarding any other.

Only articles in Portuguese or English were filtered out, as these are the languages the researchers are most familiar with in order to get a better interpretation of the articles. Another eligibility criteria for articles is that they come from one of the following websites:

- IEEE Xplore https://ieeexplore.ieee.org/Xplore/home.jsp
- ArXiv https://arxiv.org/
- SpringerLink https://link.springer.com/
- ScienceDirect https://www.sciencedirect.com/

2.2 Search strategy

Our search string included the keywords significant to the topic: “deep generative model” and “music generation”. In addition, only scientific articles from the previously time period range (2010-2023) were filtered, as well as the studies disciplines based on the proposed review objectives. That is, to search for articles dealing with the impact of artificial music generation, filters in the social sciences were used, while for the exposition of the topic, articles focusing on computer science were selected. In the case of IEEE Xplore, the search string was:

(“Full Text & Metadata”: Deep Generative models) AND (“Full Text & Metadata”: Music Generator)

Narrowing down the date of publication was another filter implemented. For the remaining sites, the advanced search was minimally modified according to the syntax and resources available to each search engine. After applying this query, the results sorted by importance were subjected to further filtering: the titles and keywords of each article, as well as their abstract, were reviewed and selected by the researchers based on their relevance to the topic of this article.

2.3 Data collection process

In order to gain insight into the research topics presented earlier, some metadata such as title, digital object identifier, year and place of publication, type of institution, type of paper and abstrat, were extracted from the articles. As we are analysing articles describing a generating music system, additional information is provided beyond the standard publication details previously mentioned:

- **Architectures**: Generative Adversarial Network (GAN), Long Short-Term Memory (LSTM), Variational Autoencoder (VAE), Transformers and others.
- **Results evaluation**: Analysis on how many studies deal with evaluation metrics and how the music produced was evaluated (objectively or subjectively).
- **Types of music**: Homophonic, heterophonic, a cappella, solo, choral, orchestral, electronic, chamber, and others.
- **Ethical considerations**: Related to the use of the generated music system, such as copyright infringement or potential harm to human users.
- **Music representation**: In the generative music system described in the paper, music can be either represented as a sequence of discrete symbols, or as a continuous waveform (an audio).
- **Dataset**: Collection of data used to train the generative music system.

At this stage, given these immense amount of information, we have developed a Python script to automate this task, based on a machine learning model that extracts sentences from articles according to the keywords presented in the list above. This script uses a SpaCy natural language processing library.

The excerpt in the appendix section A corresponds to a list of sentences for each keyword from a given article, generated in a JSON file. The idea is to run it not only for one but for all the articles selected so far to save in a file all the sentences extracted respecting those keywords. The selected articles were divided among the reviewers, each responsible for running the script for the submitted articles. After that, the reviewers analysed the document to manually gather more information directly from the articles for the "low-information" sentences.

2.4 Synthesis methods

For quantitative synthesis, we considered a visual representation of the study data. All studies selected by the reviewers will be available for quantitative analysis of the defined outcomes. In case there is an defined outcome for which the study has no data, this study will not be considered for this quantitative analysis, but will remain available for other analyses. In this way, we can weigh what should be summarize, without discarding articles for missing information. Alternatively, we can take a more qualitative approach, in which the studies selected for this analysis were filtered based on an algorithm that assigned weights to the outcomes defined in the articles. However, this algorithm only served as a starting point for recommendations to the reviewers, who were thus given the opportunity to analyze in more detail the articles they considered most relevant.

The data preparation methods for the synthesis were based on Python scripts from the information collection mentioned in section 2.3. The synthesis can be structured in both tables and bar charts. The bar charts are used to show some distributions by certain document metadata, namely the number of documents by publication year and location. In contrast, the tables are used to summarize each document selected according to the defined results.

3 Results

3.1 Study Selection

The diagram in Figure 1 indicates the number of studies filtered in the search, selection and synthesis processes for each database.

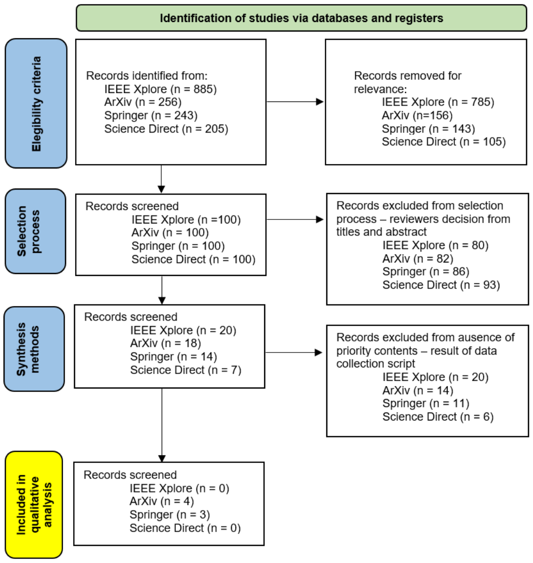
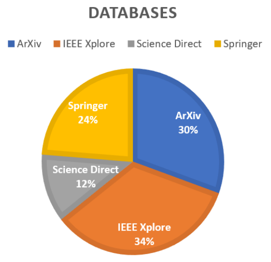
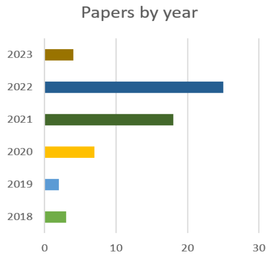


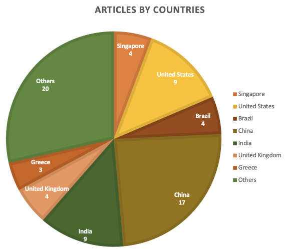
Fig. 1: Flow Diagram

3.2 Metadata analysis

The graphs down below show the distribution of metadata of the selected articles.

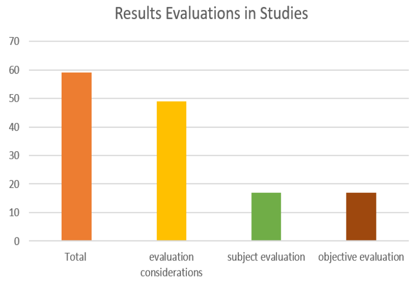


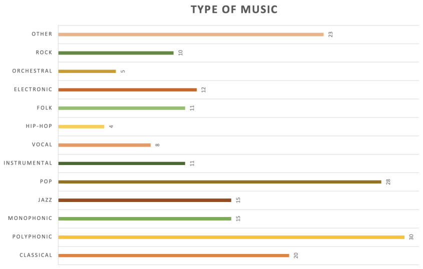




As for the distribution of datasets, a larger number of articles were selected in IEEE Xplore, but they do not differ significantly from the rest. It is interesting to see that, even considering studies from 2010, the filtered articles start in 2018 and increase significantly until 2022. Since 2023 is still very early, there are not many studies yet, but it is expected that this value will continue to increase. In addition, we can see that Asian countries, especially China and India, are strongly represented, but there is also significant research articles from the Western region, such as the United States and Europe.

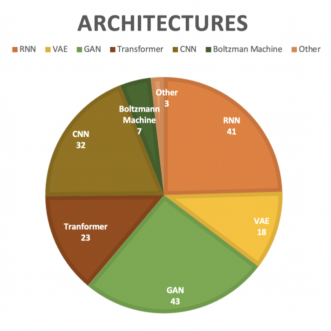
3.3 Study content analysis

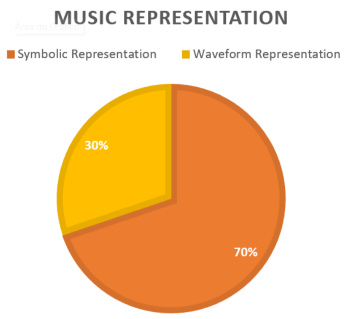




The vast majority of articles dealt with the evaluation of the produced music. However, it is interesting to note that the percentage of studies addressing subjective evaluations (of people giving their opinions while listening to the songs) is similar to the objective methods of analysing the production quality. It is often mentioned that there is no consensus or concrete forms of objective evaluation, and when they are cited, they vary greatly from one article to another.

Another analysis we can make is that a wide range of music genres are examined. Polyphonic music seems to be the most frequently studied genre, followed by pop and classical. There are fewer articles on some genres labeled as "other", such as punk, reggae, country and disco. It is noteworthy that some niche genres such as homophonic music, ambient, and opera were also studied.





Deep learning architectures such as GANs, RNNs, and CNNs have had a major impact because of their ability to generate complex and diverse designs, optimize construction, and support decision-making. GANs, in particular, are a widely studied architecture in this field. LSTM is a variation of RNN that has also contributed significantly to deep learning architectures. As these architectures evolve and evolve, we can expect more applications of architectural innovations. Although other architectures such as VAEs and Transformers were examined, their occurrence was relatively low. However, as the knowledge of their potential applications increases, they may become more relevant in the future.

As for the representation of music, waveforms are discussed in less than half as many articles as symbolic representations (e.g., MIDI, piano rolls, sheet music). This does not necessarily imply that such representation is better. These models are sometimes characterized by a lack of expressiveness [2], while the solutions with waveforms are said to have not yet been extensively researched.

3.4 Individual characteristics of some selected studies

The table 1 presents the selected studies for a more detailed individual analysis of their characteristics.

Table 1: Summary table of select studies for characteristics analysis

Study	**Architecture**	**Evaluation criteria**	**Generation**
Zhe Zhang et al. [3]	Conditional hybrid GAN	Obj[a]/Subj[b]	SMN[c] - Voice
Angela Fan et al. [2]	CycleGAN	Subj[b]	Drums
M. Conner et al. [4]	LSTM	Subj[b]	Instrumentation
Rütte et al. [5]	Vector quantized variational autoencoder	Obj[a]/Subj[b]	SMN[c]
Nathan et al. [6]	Transformer	Obj[a]/Subj[b]	SMN[c] - Piano
Fan Liu et al. [7]	$\mathrm{M}^2S-GAN^{\mathrm{d}}$	Obj[a]	Conduction motion
Xu Tan et al. [8]	Tranformer	Obj[a]/Subj[b]	SMN[c] - Piano

[a] Objective methods [b] Subjective methods

[c] Sequence of Music Notes (melody)

[d] Music Motion Synchronized Generative Adversarial Network

GAN architectures Zhe Zhang et al. [3] and Angela Fan et al. [2] were chosen because they use the architecture of GAN (which has been used most often in this context so far) in a very different way, adapted for the purpose of the study. In [3] case, the goal is to create a drum set with a bass baseline. They formulated this task as an unpaired image-to-image translation problem, starting from the conversion of the bass waveform into a mel spectrogram. The result is the generation of an original drum set that follows the beat. To achieve these results, two GAN's are needed, one for the conversion of bass lines to drums and another for the reverse conversion. The GAN's are also updated using cycle consistency losses. This is to promote the synthesized images in the target domain, which are translations of the input image (to be able to map the translation of a bass line to the corresponding drum). The goal of [2], on the other hand, is to generate melodies from song lyrics. For this purpose, a Conditioned GAN is used, which is precisely conditioned by the song lyrics. Moreover, a hybrid structure is proposed that includes three independent branches (each for a melody attribute) in the generator and a branch to discriminate chained attributes in the discriminator. Finally, a relational memory is used to model not only the dependency within each attribute sequence during the training of the generator, but also the consistency between three attribute sequences during the training of the discriminator.

Regarding the criteria for evaluating the results obtained, [3] trained a logistic regression model with features obtained by comparing the original and artificial drums. This suggests an objective evaluation, without human opinion. However, as the article itself states, the training of this network is based on quantitative measures, the results of which correlate well with (i.e., predict) the average scores of the experts. The result is still somewhat subjective and based on the experts' opinions. Therefore, it can be said that the proposal here is more for "automating" the evaluation process and not for "objectifying" it. Alternatively, [2] use objective evaluations, such as the Self- BLEU Score as a metric to measure the diversity of melodies generated and the MMD-unbiased estimator to examine the quality of the generated melodies. As can be seen, such objective measurements are more about "statistical" measures based on the success of the data distribution rather than the music itself. A subjective evaluation was also made by asking people with no knowledge of music to listen to these melodies.

Another study analysed covers the automatic generation of musical motion which are an essential component of orchestral performances. Consider the article [7] that describes M^{2}S-GAN, a neural network technique for generating conducting motion from music. It addresses the difficulties of expressing both basic beat information and articulatory information, as well as hints to different parts of the orchestra and information about the emotion expressed by the music. A competent conductor should understand the music and convey it elegantly and precisely through body movement. Three neural networks were created to do this: a music encoder, a generator, and a discriminator. The encoder collects semantic music information, whereas the generator generates a conducting motion sequence. The discriminator guarantees that the created motion matches visually the real motion, while the encoder ensures that it matches musically. The weights of these networks were created by combining feature space that is simultaneously connected to motion and music using a multimodal self-supervised learning approach. The article contrasts the suggested strategy with prior studies that applied self-supervised learning methods to generative problems. It was discovered that the M^{2}S-GAN method produces motion that is coordinated with the music without the requirement for manual annotations

LSTM architectures In recent years, creating music through LSTM[1] neural networks has gained popularity. As discussed in the article [4], RNNs[2] struggle to recall information from previous time steps due to the vanishing gradient and this is where LSTM networks come into play since they were developed to address this issue. In the experiments outlined in the article, MIDI files were first converted into a numerical input format, and then LSTM models were trained to predict the next note in a series based on input notes. The quality of the generated music and how well it fit the genre were two qualitative criteria that were used to evaluate the models. Other quantitative measures included were mean squared error and accuracy. The best-performing models were able to produce high-quality music that captured the essence of the genre and contained interesting and complex musical patterns, according to the researchers, who developed a number of models with different hyperparameters. The researchers stated that the models might still be improved and that further study is required to properly comprehend the capabilities and limitations of LSTM networks in music generation.

Transformer architectures The [6] article was chosen because it treats the production of music according to the same principles as natural language processing. In it, music is highlighted and understood as a "language" based on the following characteristics: musical notes have a temporal or natural order; just as the position of words in a sentence can determine their context, the same can happen with musical notes - any note can be related to any other, regardless of distance, and for this it is necessary to reduce the problem of long-term memory loss. Therefore, such a study is based on the Transformer architecture (similar to GPT-2 [9] and BERT [10]). The crucial point of this study is to propose a solution to the bottleneck of these networks: the temporal and spatial complexity, which grows quadratically with the length of the input sequence. Byte-Pair-Encoding is used, a compression technique. In the words of the article: "In symbolic music, notes are represented by successions of tokens that represent the values of their attributes. In this context, BPE can allow to represent a note, or even a succession of notes, that is very recurrent in the dataset, as a single token". From the proposed compression technique, despite the improvement in efficiency and training time, the tradeoff of choosing the vocabulary at the moment of compression emerges: BPE naturally increases the size of the vocabulary, while reducing the overall sequence lengths. In terms of evaluation metrics, [6] proposes automatic and objective metrics regarding the feature similarity of the generated data in relation to existing ones and based on tokenization syntax error (for example, when a NoteOff token is predicted but the corresponding note has not been played). Again it is mentioned that generative models are often evaluated with automatic metrics on the generated results, but based on data distribution. On the other hand, automatic evaluation of symbolic music remains an open issue. For this reason, a subjective method is also used based on the

[1] Long Short Term Memory

[2] Recurrent Neural Network

evaluation of the sequences generated (converted in MIDI files that reproduces a piano because the learning database had piano tracks) by three musicians.

The Transformer architecture is also used in the article [8]. Here, the structure of the song is taken into account, not just its content. That is, unlike most models that follow an end-to-end left-to-right note-by-note generative paradigm and treat each note the same, [8] incorporates the melodic skeleton as a deep structural support to provide explicit guidelines for the developmental direction of melody generation. In other words, they first generate the structurally most important notes to construct a melodic skeleton, and then fill it out with dynamically decorative notes to form a full-fledged melody. This method can improve the quality of songs, since end-to-end and data-driven learning paradigms can lead to repetitive or boring songs. The beauty of all this is that the hierarchical structure generated is from the perspective of the structure and prolongation principle. These principles are based on human creativity in music composition, where the entire process of melody creation can be viewed as a step-by-step filling in of individual decorative notes within the melodic skeleton. This article therefore differs from all the others analyzed because it discusses the element of “creativity” and attempts to explain concepts from music theory. The experimental results of this approach outperformed other five current SOTA[3] models for end-to-end left-to-right note-by-note melody generation in all metrics, including (1) MusicTransformer, (2) Pop Music Transformer, (3) Compound Word Transformer, (4) Melons, and (5) MeMIDI. The results were compared in terms of objective metrics by calculating the average overlap area of some musical feature distributions between generated music tracks and real music tracks. However, it is mentioned again that there are no convincing and unified objective metrics because it is difficult to compare different music generation systems, as the research field of music generation is multidisciplinary with different methods and goals. Therefore, subjective evaluation by humans is also used in this study. Finally, a major plus of this approach is that it can not only maintain the long-range tonal coherence of the generated melodies, but also achieve control over the target of melodic motion by human users.

Other architectures The following article [5] discussed how symbolic music may be generated with fine-grained control using a neural network model. The primary goals are to create a descriptive conditioning sequence and train a generative model using the data from that sequence. The expert description, which is created using domain knowledge and is interpretable by humans, and the learned description, which is produced using the VQ-VAE framework, are the two techniques this article offers for generating descriptions from musical sequences. Both methods produce high-quality music with fine-grained control, making them more effective than global control. Fine-grained control is possible with the neural network model used in the study, which might help with the building of more advanced music production tools. The researchers also ran tests

[3] State Of The Art

to evaluate the way the models performed in different activities. Both models produce high-quality music with fine-grained control, according to the results of the first experiment, conditional generation. On most metrics, the learned description model performed better than Choi et al. (2020) [11]. The results of the second experiment, using zero-shot medley generation, demonstrated that the models which employed the expert description as input outperformed FIGARO (the learned one) and Choi et al (2020). Additionally, FIGARO outperformed every other baseline in sample quality while offering controlled production capabilities, in line with a user survey that also assessed the subjective quality of generated samples.

4 Discussion

From the results shown, we can conclude that the music generation models based on GAN, LSTM and Transformers are the most widely used nowadays. In terms of music representation, we found a significant superiority of symbolic representation compared to non-symbolic representation (e.g., waveforms). Regarding evaluation metrics, there is a relevant gap in objective metrics. Therefore, most studies propose both subjective scoring (to analyze the quality of the results) and objective scoring, which provides information on how similar the result produced is to the existing music.

Due to the massive data analysis in the information collection phase (about 59 articles), automation in Python was proposed. However, this strategy is tightly constrained by the search keywords defined. Moreover, we know that the detection of a keyword does not mean that the article necessarily covers that topic. Therefore, we are aware of the limitations of automating this process and thus we always try to manually verify the results obtained.

Having reached this point, the absence of the subject of ethical concerns in the studies of authorship and rights to the music produced is easily noticed. This was not unintentional, however, for in fact practically no study has addressed such a topic. In terms of a more human analysis of ”creativity” and discussion of the related concept of “art” only study [8] quickly addressed it. Thus, for future work in this area, it is recommended to build models that take these factors into account, and not just worry about producing any results.

In summary, we conclude that the generation of music by Deep Learning is still at an early stage, without a great consensus in the evaluation of the generated music and in a global and protocol process of generation capable of capturing the essence of what is considered art by humans.

5 Other Information

Reviewer-selected articles, data collection scripts, and data extracted for study summaries are collected in the Git repository Music-Generation.

References

1. Big Data Framework. A Short History of Big Data, 2021. Accessed on March 20, 2023.
2. Giorgio Barnabò, Giovanni Trappolini, Lorenzo Lastilla, Cesare Campagnano, Angela Fan, Fabio Petroni, and Fabrizio Silvestri. CycleDRUMS: Automatic Drum Arrangement For Bass Lines Using CycleGAN, 2021.
3. Wei Duan Abhishek Srivastava Rajiv Shah Yi Ren Yi Yu, Zhe Zhang. Conditional hybrid GAN for melody generation from lyrics, 2022.
4. Michael Conner, Lucas Gral, Kevin Adams David Hunger, Reagan Strelow, and Alexander Neuwirth. Music Generation Using an LSTM, 2022.
5. Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. FIGARO: Generating Symbolic Music with Fine-Grained Artistic Control, 2022.
6. Nathan Fradet, Jean-Pierre Briot, Fabien Chhel, Amal El Fallah Seghrouchni, and Nicolas Gutowski. Byte Pair Encoding for Symbolic Music, 2023.
7. Fan Liu, De-Long Chen, Rui-Zhi Zhou, Sai Yang, and Feng Xu. Self-Supervised Music Motion Synchronization Learning for Music-Driven Conducting Motion Generation, 2022.
8. Kejun Zhang, Xinda Wu, Tieyao Zhang, Zhijie Huang, Xu Tan, Qihao Liang, Songruoyao Wu, and Lingyun Sun. WuYun: Exploring hierarchical skeleton-guided melody generation using knowledge-enhanced deep learning.
9. Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Better Language Models and Their Implications. *OpenAI blog*, 1(8):9, 2019.
10. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
11. Kristy Choi, Curtis Hawthorne, Ian Simon, Monica Dinculescu, and Jesse Engel. Encoding Musical Style with Transformer Autoencoders. *Proceedings of the 37th International Conference on Machine Learning*, 119, 2020.

A Data Collection Script

Listing 1.1: data collection script

```

def generateDataCollection(keywords, output):
    nlp = spacy.load("en_core_web_sm")

    # Result Json
    result = {}
    result["data_collection"] = []

    # getAllMetadata returns pdf filenames to read
    allMetaData = getAllMetaData()

    # Language processing in each file
    i = 0
    for metadata in allMetaData["content"]:
        reader = PdfReader(metadata["database"] + "/" +
            metadata["file"])
        content = ''
        # merge ao pdf content in one variable
        for page in reader.pages:
            content += page.extract_text()

        doc = nlp(content)
        articleData = {}
        articleData["title"] = metadata["title"]

        for keyword in keywords:
            articleData[keyword] = []

        # detect which sentence has the keywords definied
        # and add to article keyword entry
        for sent in doc.sents:
            for keyword in keywords:
                if keyword in sent.text.strip():
                    articleData[keyword].append(sent.text.
                        strip())

        result["data_collection"].append(articleData)

        print("[" + str(i) + "]␣Article␣'" + metadata["
            database"]  + "/" + metadata["file"] + "'␣
            processed")
        i = i + 1

    with open(output + ".json", 'w') as f:
        json.dump(result, f, indent=4)
```