# From Statistics to Neural Networks: Enhancing Ciphertext-Plaintext Alignment in Historical Substitution Ciphers for Automatic Key Extraction

**Micaella Bruton**
Stockholm University
micaella.bruton@ling.su.se

**Beáta Megyesi**
Stockholm University
beata.megyesi@ling.su.se

## Abstract

Ciphertext manuscripts found in archival collections are often intermingled with plaintext manuscripts in various languages, making the manual analysis required to match the documents labour-intensive and complex. Automating the alignment of these texts to reconstruct corresponding cipher keys is therefore highly beneficial, particularly when handling large volumes of documents. This study introduces a novel approach using modern neural networks, specifically Long Short-Term Memory (LSTM) architectures, to develop an automated method for aligning homophonic substitution ciphertexts with plaintext. These neural models are compared to traditional statistical approaches, demonstrating that LSTMs achieve significant accuracy improvements, including perfect alignment for ciphertexts of 50 characters or less. Additionally, to facilitate practical application, a program has been developed to enable the upload of transcribed ciphertext and plaintext documents, using the optimized models to automatically align the texts and extract the substitution key.

## 1 Introduction

National libraries and archives worldwide house collections containing ciphertexts and cipher keys alongside manuscripts written in plaintext. Cryptanalysts frequently begin their work by attempting to align ciphertext with plaintext to recover the underlying cipher key. This process is inherently complex and time-consuming, underscoring the value of automation. Computational methods, whether statistical or neural, offer significant potential to streamline this process.

One prominent example is the DECODE database, which contains 3,285 ciphertexts and 5,717 keys, yet only 13.9% have been fully decrypted or matched to their respective keys (Héder and Megyesi, 2022; Megyesi et al., 2019; Megyesi et al., 2020). The decryption of such texts remains labor-intensive due to the scarcity of contextual information and the complexity of historical cipher systems. These challenges underscore the necessity for advanced computational tools to complement manual methods and accelerate cryptanalysis.

Modern computational methods, particularly neural networks, offer unprecedented opportunities to address these challenges. By leveraging their ability to process sequential data and identify complex patterns, neural networks can assist in systematically aligning ciphertext with plaintext. Text alignment is a crucial step in systematic decryption, enabling cryptanalysts to reconstruct encrypted messages, extract keys, and gain insights into historical communications.

This paper explores the application of Long Short-Term Memory (LSTM) models for aligning ciphertext and plaintext in historical homophonic substitution ciphers. LSTMs are particularly well-suited for this task due to their ability to capture long-range dependencies in sequential data. By evaluating the efficacy of these models and comparing them to traditional statistical approaches, this study aims to bridge the gap between modern computational advancements and the unique challenges posed by historical cryptanalysis.

### Contributions

The primary contributions of this work are as follows:

- to evaluate the performance of LSTM models in character-level alignment for historical ciphertext decryption;

*Proceedings of the 8th International Conference on Historical Cryptology*, pages 18-29
Poznań, Poland, 16-18 June, 2025

- to investigate the extent to which neural models reduce the reliance on preprocessing techniques, such as chunking, which involves dividing text into smaller segments to facilitate analysis and processing, often at the cost of losing contextual information and introducing potential misalignments in the data;
- to assess the impact of incorporating synthetic data on improving model generalization and performance on historical datasets;
- and, to develop and release a user-friendly program that enables individuals without coding expertise to perform automatic text alignment and key extraction methods on their personal computer.

## 2 Background

### 2.1 Text Alignment

Text alignment has long been a foundational task in Natural Language Processing (NLP), commonly used to align tokens and sentences across languages for applications such as machine translation, information retrieval, text entailment, and question answering (Oakes and McEnery, 2000; Véronis and Langlais, 2000; Zha et al., 2024; Bahdanau, 2014; Semmar and Fluhr, 2007). In the context of cryptanalysis, text alignment is equally critical, aiding in the decipherment of unknown or partially understood ciphers. This process involves matching sequences of ciphertext to potential plaintext equivalents by comparing encrypted text to known patterns, linguistic structures, or historical texts.

Unlike typical NLP tasks, where white-space often serves as a natural delimiter for tokens, historical ciphertexts frequently lack such boundaries. The removal of white-space complicates tokenization, as current systems predominantly rely on tokens or words for alignment. Without clear separations, the task shifts from aligning distinct tokens to aligning continuous sequences, a significantly more complex undertaking. This challenge is often further compounded as code groups in the ciphertext symbol sequence may vary in length, often comprising 2-, 3-, or 4-digit codes (Megyesi et al., 2024). Such variability adds another layer of difficulty, as decipherment systems must accurately segment and align symbols of different lengths without prior knowledge of their boundaries.

Table 1 provides an example of aligned text, illustrating multiple ciphertext characters—represented here as double-digit numbers—are mapped to the same plaintext character as an added security measure. This practice, known as homophonic substitution, underscores the importance of text alignment in detecting recurring patterns and facilitating systematic decryption. By addressing these challenges, advanced alignment techniques can enable cryptanalysts to reconstruct cipher keys and derive meaningful insights from encrypted communications.

| ... | 58 | **92** | 33 | **23** | 32 | 96 | 37 | **92** | **28** | 91 | **19** | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | A | **M** | O | **N** | G | W | Y | **M** | **M** | E | **N** | ... |

Table 1: Aligned Text Showcasing Homophonicity

### 2.2 Traditional Cryptanalysis & Statistical Approaches

Cryptanalysis has traditionally relied on manual techniques such as frequency analysis, pattern recognition, and linguistic analysis. These methods, while effective in solving simple substitution ciphers, face significant challenges when applied to more complex encryption systems, such as homophonic ciphers or those employing multiple alphabets (Megyesi et al., 2023a). The complexity of such systems significantly increases the effort required for decryption, often making manual methods impractical for large-scale or highly intricate cryptographic challenges.

A notable example is the Zodiac Killer's cipher, whose decryption required over 50 years to complete, even with significant advancements in computational power and collaborative efforts (Oranchak et al., 2024). This highlights the limitations of traditional cryptanalytic methods and underscores the necessity of integrating modern computational techniques to address these challenges more efficiently.

Statistical models have been used to automate parts of the text alignment process. IBM alignment models, originally developed for automatic machine translation, have been widely used used for years and have achieved high accuracy in text alignment tasks, but require significant preprocessing (Brown et al., 1993; Och and Ney, 2003; Koehn et al., 2003; Boglind, 2024). Boglind (2024) evaluated statistical IBM models on ciphertext/plaintext alignment for texts up to character length 2400. Though the best results reported

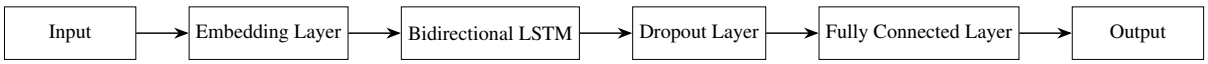


Figure 1: Model Architecture, Illustrating the Flow from Input Sequences to Output Predictions.

near-perfect alignment, this performance relied on segmenting the text into smaller segments, a process known as chunking (Boglind, 2024). While chunking enhances alignment accuracy, it can introduce significant computational overhead, particularly when applied to large-scale datasets or resource-constrained environments, potentially limiting its feasibility for real-time processing. Additionally chunking increases the risk of misalignment by reducing contextual information, which is crucial for accurate alignment. The choice of chunk length also has a substantial impact on performance; for example, increasing the chunk size from 5 to 40 resulted in a more than 70% decrease in the *F1* score; these baseline results can be found in Figure 2 (Boglind, 2024). While these statistical methods are effective when applied to preprocessed and annotated data, their utility is limited in the context of historical ciphers, which often lack tokenized and annotated data.

Despite their utility, these statistical methods face limitations in addressing the unique challenges of historical ciphers. Historical texts often lack token boundaries, exhibit irregular symbol usage, and are prone to transcription errors (Megyesi et al., 2023a). These irregularities complicate alignment tasks, rendering statistical models less effective, as they rely on clearly defined patterns and annotated data. This highlights the need for more adaptive methods to overcome these challenges.

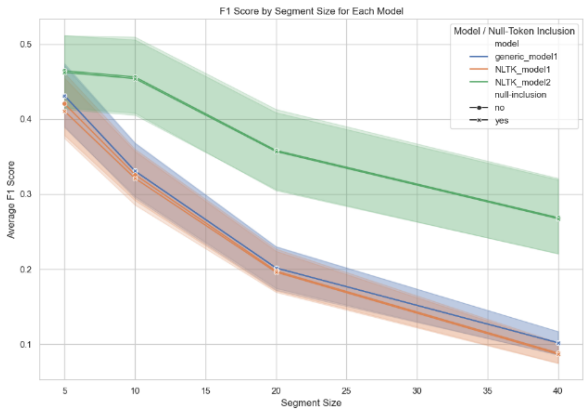


Figure 2: Boglind (2024)'s Results - *F1* Score for Segment (Chunk) Size

### 2.3 Neural Approaches in Cryptanalysis

Neural networks have demonstrated significant potential addressing the limitations of traditional methods. Sequence-to-sequence models, such as LSTMs, excel at processing sequential data and capturing long-range dependencies, making them well-suited for tasks like text alignment. Applications of neural networks to cryptanalysis have largely focused on image-to-text alignment or transcription tasks, however, their application to ciphertext/plaintext alignment remains underexplored (Torras et al., 2021; Fischer et al., 2011; Torras et al., 2023).

Neural models are well-equipped to address many of the additional challenges presented by historical ciphers (Megyesi et al., 2023a). Unlike traditional statistical models, neural models offer greater flexibility and adaptability in regards to preprocessing and annotated data, particularly when trained on both real-world and synthetic datasets, allowing them to generalize and adapt to complex alignment tasks. Synthetic datasets, which are generated computationally to simulate realistic patterns found in historical ciphers, are particularly valuable in augmenting training data when real-world examples are limited. By incorporating both types of data, neural models can better handle the irregularities and ambiguities inherent in historical cryptanalysis.

Recent studies have highlighted the potential of neural methods for historical cryptology. For example, Megyesi et al. (2023b) demonstrated that language models trained on century-specific historical texts improved the decryption of homophonic substitution ciphers. These findings underscore the importance of incorporating historical context into neural models to enhance their performance on historical datasets.

## 3 Method

To align each ciphertext character with its plaintext equivalent, models were trained on homophonically encrypted texts using full sequences of fixed lengths—50-, 100-, and 200-

| Text Length | Train Data | Embedding Dim | Hidden Size | Layers | Dropout | Learning Rate | Batch Size |
|---|---|---|---|---|---|---|---|
| 50 | Real-World | 109 | 196 | 3 | $2.97 \times 10^{-1}$ | $5.45 \times 10^{-4}$ | 59 samples |
| | Synthetic | 246 | 104 | 3 | $3.56 \times 10^{-1}$ | $3.77 \times 10^{-4}$ | 49 samples |
| | Joint | 190 | 225 | 3 | $3.36 \times 10^{-1}$ | $6.40 \times 10^{-4}$ | 43 samples |
| 100 | Real-World | 204 | 247 | 3 | $3.42 \times 10^{-1}$ | $1.34 \times 10^{-3}$ | 19 samples |
| | Synthetic | 149 | 253 | 3 | $3.83 \times 10^{-1}$ | $1.27 \times 10^{-3}$ | 37 samples |
| | Joint | 264 | 269 | 3 | $4.47 \times 10^{-1}$ | $8.04 \times 10^{-4}$ | 24 samples |
| 200 | Real-World | 117 | 184 | 2 | $3.26 \times 10^{-1}$ | $1.70 \times 10^{-3}$ | 20 samples |
| | Synthetic | 215 | 291 | 3 | $3.33 \times 10^{-1}$ | $6.37 \times 10^{-4}$ | 27 samples |
| | Joint | 149 | 263 | 2 | $3.32 \times 10^{-1}$ | $1.45 \times 10^{-3}$ | 26 samples |

Table 2: Hyperparameters for each model, as defined by Optuna (Akiba et al., 2019)

characters—without chunking, in order to maintain the integrity of the input data. This approach was designed to enable the models to learn both local and global patterns within the ciphertext, such as symbol-to-character correspondences, recurring symbol groupings, contextual alignments across sequences, and the statistical distribution of homophonic substitutions.

Three models were trained for each length using distinct datasets: ciphertext/plaintext pairs derived from real-world historical plaintext data, ciphertext/plaintext pairs derived from synthetic historical plaintext data, and a combination of both data types. Synthetic datasets were incorporated to measure their impact on model performance, providing a means to supplement the limited availability of historical data while exposing the models to diverse patterns and structures that replicate the characteristics of historical cipher texts.

The evaluation process involved testing the models across varying ciphertext-to-plaintext mapping ratios, ranging from 1:1 to 5:1. These ratios simulate the complexity of real-world historical ciphers. By including these varied mappings, the study assesses the models' ability to adapt to increasing levels of homophonic complexity and maintain alignment accuracy under diverse cryptographic conditions.

The project repository is available on GitHub[1].

### 3.1 Data

To train and evaluate the models, datasets were constructed using 35,000 English plaintexts, each with a length of 200 characters. Of these, 1000 texts were reserved for both the validation and test sets, ensuring that training, validation, and testing data remained distinct. For consistency, the same splits were used when training models on the 50- and 100-character sequence lengths, allowing for a direct comparison of performance across different sequence lengths.

#### Plaintext

Plaintext data was sourced from the HistCorp and Project Gutenberg corpora, restricted to English texts spanning the years 1350–1899 (Pettersson and Megyesi, 2018; Gerlach and Font-Clos, 2020). Synthetic plaintext sequences were generated using 5-gram models trained on these corpora. To prevent overfitting or memorization, all generated sequences were ensured to be unique and distinct from the original data. Post-generation, the datasets were inspected to verify the absence of duplicates. For joint datasets, an equal distribution of real-world and generated plaintexts was maintained to ensure balanced representation.

#### Ciphertext

The Python library ChronoFidelius[2] was developed as part of this work and used to homophonically encrypt all plaintexts. Each plaintext character was mapped to between 1 and 5 unique four-digit numbers. For the training datasets, the mapping was weighted by historical character frequencies to emulate realistic plaintext distributions, ensuring that the models did not benefit from frequency imbalances. For example, high-frequency characters were mapped to one of five numbers, while low-frequency characters were assigned a single number.

The evaluation datasets were encrypted using flat mapping ratios ranging from 1:1 to 5:1, defined as ciphertext_character:plaintext_character. Spaces were excluded to replicate the characteristics of real-world historical ciphertexts. Additionally, no transcription errors or noise were in-

[1] https://github.com/mbruton0426/ciphertext-plaintext-alignment

[2] https://github.com/mbruton0426/ChronoFidelius

troduced, ensuring the evaluation focused exclusively on the model's alignment capabilities under ideal conditions.

### 3.2 Model

The model architecture is shown in Figure 1. A bidirectional LSTM captures dependencies from both preceding and succeeding tokens, enhancing its ability to identify alignment patterns. This is followed by a dropout layer to mitigate overfitting, and a fully connected layer that outputs logits for classification. Training was performed using the AdamW optimizer and a Cross-Entropy loss function, with Xavier uniform initialization to promote weight stability (Kingma, 2014; Glorot and Bengio, 2010). Training was conducted on a single NVIDIA A40 GPU until convergence, stopping after five epochs without improvement in validation loss.

Hyperparameter optimization was performed using Optuna, with 50 trials evaluated to determine the best-performing configuration based on validation set performance. The TPESampler and MedianPruner were utilized to improve optimization efficiency (Akiba et al., 2019). Table 2 details the final hyperparameter configurations for each model.

### 3.3 Evaluation

Performance was assessed using accuracy and the *F1* score to evaluate the model's character-level alignment capabilities. These metrics account for both false positives and false negatives, which are critical in cryptographic applications. They are defined as follows:

- **Accuracy**: The ratio of correct character-level predictions to total character-level predictions;
- ***F1* Score**: The harmonic mean of precision; defined as $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$, where *precision* measures the proportion of correctly predicted positive instances out of all predicted positive instances, and *recall* measures the proportion of correctly predicted positive instances out of all actual positive instances. This metric provides a balanced assessment of model performance.

## 4 Results

Due to the large number of evaluation sets, complete results are provided in Appendices A, B, and C for models trained on 50-, 100-, and 200-character sequences respectively. Overall, shorter input sequences and lower ciphertext-to-plaintext mapping options led to improved performance. Models trained on a combination of real-world and synthetic data consistently achieved the best results on texts longer than 50 characters.

Key performance highlights are summarized below, with a primary focus on results obtained from real-world test data, as the ultimate goal of this study is to improve alignment and decryption performance on historical ciphers. The highest accuracy and *F1* score achieved for real-world test data were as follows:

- **50-character sequences** 100% accuracy and *F1* score
- **100-character sequences** 67.8% accuracy and 69.24% *F1* score
- **200-character sequences** 52.16% accuracy and 53.36% *F1* score

### 4.1 Text Length 50

Models trained on 50-character sequences demonstrated exceptional performance across all data types and key configurations. Notably, models trained exclusively on either real-world or synthetic data achieved perfect classification scores on real-world test data, indicating flawless alignment.

Models trained on a combination of real-world and synthetic data showed a marginal decrease in performance, though they maintained an accuracy and *F1* score over 90% in all cases.

### 4.2 Text Length 100

For models trained on 100-character sequences, performance declined slightly across all conditions compared to shorter sequences. However, real-world test data generally remained the highest-performing category, with models trained on a combination of real-world and synthetic data achieving the best results in this setting.

Models trained exclusively on synthetic data achieved the highest overall scores, with an accuracy of 83.91% and an *F1* score of 84.60%. In contrast, models trained solely on real-world data showed significant limitations, with top accuracy and *F1* scores of 47.28% and 47.46%, respectively. Models trained on both real-world and

synthetic data demonstrated superior generalization compared to real-world-only models, achieving top scores of 67.79% accuracy and 69.24% *F1*.

### 4.3 Text Length 200

As expected, models trained on sequences of length 200 faced the greatest challenges. Models trained on both real-world and synthetic data achieved performance comparable to real-world-only models trained on 100-character, with a peak accuracy of 52.58% and an *F1* score of 54.14%.

Models trained exclusively on synthetic data struggled to generalize to real-world data, resulting in consistently low scores across all metrics. Among models trained exclusively on real-world data, the highest-performing achieved an accuracy of 35.97% and an *F1* score of 37.11%.

### 4.4 Program Implementation

A stand-alone program will be released and linked to the project repository, operationalizing the best-performing models. This application allows users to upload two text documents, one representing ciphertext and the other plaintext, after which the program automatically aligns the texts and extracts the corresponding substitution key. Both the aligned text and the extracted key are returned as text documents.

This program offers an open-source, accessible tool for historical cryptanalysts. By providing a user-friendly interface, it eliminates the need for coding or advanced computational skills, making automated text alignment and key extraction more widely accessible.

## 5 Discussion

This study reveals a consistent pattern: LSTM models trained on shorter input sequences, particularly those incorporating a mix of real-world and synthetic data, achieve superior alignment performance across all configurations. In contrast, longer sequences pose significant challenges, especially for models trained exclusively on real-world data.

The following sections examine the implications of these findings for cipher alignment and decryption tasks and propose recommendations for future research to address the identified limitations.

### 5.1 Performance Trends by Text Length

#### Text Length 50

The highest alignment accuracy was consistently achieved by models trained on 50-character sequences. Configurations trained exclusively on either real-world or synthetic data reached perfect alignment scores, suggesting that shorter sequences provide sufficient contextual information for accurate alignment while minimizing the complexity introduced by longer sequences.

This observation aligns with manual cryptanalysis practices, where analysts often examine short text segments to identify alignment patterns efficiently. Furthermore, these results significantly outperformed baseline statistical approaches, as the best reported statistical models required chunking text into segments of five characters to achieve best performance (Boglind, 2024).

#### Text Length 100

As the input length increased to 100 characters, model performance declined across all configurations. However, models trained on both real-world and synthetic data significantly outperformed those trained exclusively on a single data type when tested on real-world data. This suggests that incorporating synthetic plaintext data enables the model to capture a broader range of alignment patterns, improving generalization as text length increases.

#### Text Length 200

The most significant challenges were observed with 200-character sequences. While models trained on joint datasets continued to outperform all other configurations, their overall performance was still markedly lower than that of models trained on shorter sequences. This suggests that the complexity of alignment patterns in longer sequences exceeds the capacity of the LSTM architecture to generalize effectively, even with the inclusion of synthetic data.

These findings highlight a fundamental limitation of LSTMs in long-text alignment tasks and suggest the need for more advanced architectures or complementary strategies, such as attention mechanisms, to perform performance such cases.

### 5.2 Comparative Analysis of Training Data Types

Across longer text sequences, models trained on joint datasets demonstrated superior generaliza-

tion compared to those trained exclusively on either real-world or generated data. The inclusion of synthetic plaintext data in the training set appears to provide necessary diversity to capture a wider array of alignment patterns and develop robustness in variations in text structure.

In contrast, models trained solely on real-world data exhibited limited variability, which constrained their performance on longer sequences. This reinforces the importance of incorporating synthetic data during training to mitigate overfitting and improve alignment accuracy. Joint datasets provide a balanced approach by combining the specificity of real-world data with the broader coverage of synthetic data, offering a particularly effective solution for shorter sequences where alignment challenges are less pronounced.

### 5.3 Implications for Real-World Contexts & Future Work

The findings of this study highlight the critical role of input length and training data composition in developing neural models for text alignment tasks. Shorter text sequences achieve high alignment accuracy, making them particularly useful for applications involving shorter ciphers in real-world contexts. However, aligning longer texts remains a challenge, suggesting that additional pre- and post-processing techniques or varying architectures may be beneficial. While LSTM-based models demonstrate strong alignment capabilities, their effectiveness diminishes with longer sequences, indicating a need for alternative approaches.

Future research into neural methods for ciphertext/plain text alignment should prioritize the incorporation of synthetic data into training sets, as this approach enables condensed datasets to better reflect real-world text patterns. Efforts to enrich real-world datasets with greater variability should also be pursued to enhance model adaptability. Additionally, exploring architectures beyond LSTMs, such as Transformer-based models, could offer improved performance on longer sequences due to their ability to model long-range dependencies more effectively. While attention mechanisms within LSTMs could enhance performance, fully Transformer-based approaches, including BERT or sequence-to-sequence Transformer models, may be more promising alternatives. While such models can require larger amounts of training data, the inclusion of synthetic data could mitigate this requirement. Furthermore, data augmentation and domain adaptation techniques may improve model generalization to unseen real-world data, while accounting for the complexities of historical ciphertexts.

The present study was restricted to English-language data; future work should extend the approach to other languages in order to assess cross-linguistic robustness. Moreover, introducing errors which are commonly encountered in historical ciphertexts, such as transcription inaccuracies and typographical errors, would increase the practical applicability of these models for real-world cryptanalysis tasks.

## 6 Conclusion

This study investigated the influence of input sequence length and training data composition on the alignment accuracy of LSTM models designed for ciphertext/plaintext alignment tasks. The results demonstrate that models trained on shorter sequences, particularly those leveraging a combination of real-world and synthetic data, consistently achieve the highest accuracy and *F1* scores.

Synthetic data emerged as a valuable resource, enhancing generalization and enabling models to better handle diverse alignment scenarios, particularly as input length increases. The results also highlight the critical importance of optimizing both training data composition and sequence length when developing alignment models for cryptographic applications.

Despite these advancements, certain limitations remain. The study focused exclusively on English-language data and idealized inputs, and extending this approach to other languages would be necessary to assess cross-linguistic and real-world robustness. Additionally, while the models performed well on shorter sequences, their effectiveness on longer texts remains constrained. Continued exploration of architectures beyond LSTMs, particularly those capable of modeling long-range dependencies, may offer improved scalability to address this issue.

By bridging modern neural approaches with the challenges of historical cryptographic analysis, this work provides a foundation for developing robust, accessible tools that can accelerate decryption tasks and deepen our understanding of past cryptographic practices.

## Acknowledgments

The project is financed by the Swedish Research Council, partially by DECRYPT - Decryption of Historical Manuscripts (grant 2018-06074), and partially by The Swedish Graduate School of Digital Philology (grant 2022-06343). The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council (grant 2022-06725).

## References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Dzmitry Bahdanau. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Fredrik Boglind. 2024. Aligning historical ciphertext and plaintext using statistical machine translation methods. Bachelor's thesis, Department of Linguistics, Stockholm University.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Andreas Fischer, Volkmar Frinken, Alicia Fornés, and Horst Bunke. 2011. Transcription alignment of Latin manuscripts using hidden Markov models. In *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing*, pages 29–36.

Martin Gerlach and Francesc Font-Clos. 2020. A standardized project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics. *Entropy*, 22(1):126.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Mihály Héder and Beáta Megyesi. 2022. The Decode database of historical ciphers and keys: Version 2. In *International Conference on Historical Cryptology*, pages 111–114.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL 2003)*, pages 48–54. Association for Computational Linguistics.

Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The decode database: Collection of historical ciphers and keys. In *The 2nd International Conference on Historical Cryptology, HistoCrypt 2019, June 23-26 2019, Mons, Belgium*, pages 69–78.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of historical manuscripts: The DECRYPT project. *Cryptologia*, 44(6):545–559, November.

Beáta Megyesi, Alicia Fornés, Nils Kopal, Benedek Láng, Michelle Waldispühl, Vasily Mikhalev, and Bernhard Esslinger. 2023a. *Historical Cryptology*. Artech House.

Beáta Megyesi, Justyna Sikora, Filip Fornmark, Michelle Waldispühl, Nils Kopal, and Vasily Mikhalev. 2023b. Historical language models in cryptanalysis: Case studies on English and German. In *Proceedings of the 6th International Conference on Historical Cryptology HistoCrypt 2023*, pages Published on May 30, 2023. Linköping University Electronic Press.

Beáta Megyesi, Crina Tudor, Benedek Láng, Anna Lehofer, Nils Kopal, Karl de Leeuw, and Michelle Waldispühl. 2024. Keys with nomenclatures in the early modern Europe. *Cryptologia*, 48(2):97–139.

Michael Oakes and Tony McEnery. 2000. Bilingual text alignment—an overview. *Multilingual corpora in teaching and research*, pages 1–37.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

David Oranchak, Sam Blake, and Jarl Van Eycke. 2024. The solution of the Zodiac killer's 340-character cipher. *arXiv preprint arXiv:2403.17350*.

Eva Pettersson and Beáta Megyesi. 2018. The HistCorp collection of historical corpora and resources. In *DHN 2018: The Third Conference on Digital Humanities in the Nordic Countries*, pages 306–320, Helsinki, Finland. University of Helsinki.

Nasredine Semmar and Christian Fluhr. 2007. Arabic to French sentence alignment: Exploration of a cross-language information retrieval approach. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 73–80.

Pau Torras, Mohamed Ali Souibgui, Jialuo Chen, and Alicia Fornés. 2021. A transcription is all you need: Learning to align through attention. In Elisa H. Barney Smith and Umapada Pal, editors, *Document Analysis and Recognition – ICDAR 2021 Workshops*, pages 141–146, Cham. Springer International Publishing.

Pau Torras, Mohamed Ali Souibgui, Jialuo Chen, Sanket Biswas, and Alicia Fornés. 2023. Segmentation-free alignment of arbitrary symbol transcripts to images. In Mickael Coustaty and Alicia Fornés, editors, *Document Analysis and Recognition – ICDAR 2023 Workshops*, pages 83–93, Cham. Springer Nature Switzerland.

Jean Véronis and Philippe Langlais. 2000. Evaluation of parallel text alignment systems: The arcade project. *Parallel text processing: Alignment and use of translation corpora*, pages 369–388.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2024. Text alignment is an efficient unified model for massive nlp tasks. *Advances in Neural Information Processing Systems*, 36.

## A Results: Models Trained on Data Length 50

| Key Option | Train Data | Test Data | Accuracy | *F1* |
|---|---|---|---|---|
| 1:1 | Real-World | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9162 | 0.9169 |
| | | Joint | 0.9581 | 0.9582 |
| | Synthetic | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9192 | 0.9190 |
| | | Joint | 0.9596 | 0.9595 |
| | Joint | Real-World | 0.9891 | 0.9890 |
| | | Synthetic | 0.9141 | 0.9137 |
| | | Joint | 0.9516 | 0.9513 |
| 2:1 | Real-World | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9152 | 0.9159 |
| | | Joint | 0.9576 | 0.9577 |
| | Synthetic | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9181 | 0.9179 |
| | | Joint | 0.9590 | 0.9589 |
| | Joint | Real-World | 0.9897 | 0.9895 |
| | | Synthetic | 0.9135 | 0.9131 |
| | | Joint | 0.9516 | 0.9513 |
| 3:1 | Real-World | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9147 | 0.9153 |
| | | Joint | 0.9574 | 0.9575 |
| | Synthetic | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9177 | 0.9175 |
| | | Joint | 0.9589 | 0.9587 |
| | Joint | Real-World | 0.9899 | 0.9897 |
| | | Synthetic | 0.9128 | 0.9124 |
| | | Joint | 0.9513 | 0.9511 |
| 4:1 | Real-World | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9145 | 0.9152 |
| | | Joint | 0.9573 | 0.9574 |
| | Synthetic | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9175 | 0.9173 |
| | | Joint | 0.9587 | 0.9586 |
| | Joint | Real-World | 0.9895 | 0.9893 |
| | | Synthetic | 0.9128 | 0.9125 |
| | | Joint | 0.9511 | 0.9509 |
| 5:1 | Real-World | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9144 | 0.9150 |
| | | Joint | 0.9572 | 0.9573 |
| | Synthetic | Real-World | 1.0000 | 1.0000 |
| | | Synthetic | 0.9174 | 0.9172 |
| | | Joint | 0.9587 | 0.9586 |
| | Joint | Real-World | 0.9894 | 0.9892 |
| | | Synthetic | 0.9127 | 0.9123 |
| | | Joint | 0.9510 | 0.9507 |

Table 3: Performance of models trained on sequences of length 50 — Best results overall and for real-world test data are highlighted

## B Results: Models Trained on Data Length 100

| Key Option | Train Data | Test Data | Accuracy | *F1* |
|---|---|---|---|---|
| 1:1 | Real-World | Real-World | 0.4728 | 0.4755 |
| | | Synthetic | 0.4724 | 0.4747 |
| | | Joint | 0.3642 | 0.3500 |
| | Synthetic | Real-World | 0.2556 | 0.2401 |
| | | Synthetic | 0.5768 | 0.6078 |
| | | Joint | 0.7078 | 0.7220 |
| | Joint | Real-World | 0.6779 | 0.6924 |
| | | Synthetic | 0.6209 | 0.6330 |
| | | Joint | 0.7075 | 0.7211 |
| 2:1 | Real-World | Real-World | 0.4724 | 0.4746 |
| | | Synthetic | 0.4723 | 0.4736 |
| | | Joint | 0.3640 | 0.3497 |
| | Synthetic | Real-World | 0.2560 | 0.2415 |
| | | Synthetic | 0.5761 | 0.6067 |
| | | Joint | 0.7073 | 0.7205 |
| | Joint | Real-World | 0.6777 | 0.6885 |
| | | Synthetic | 0.6207 | 0.6314 |
| | | Joint | 0.7073 | 0.7202 |
| 3:1 | Real-World | Real-World | 0.4724 | 0.4738 |
| | | Synthetic | 0.4724 | 0.4746 |
| | | Joint | 0.3642 | 0.3500 |
| | Synthetic | Real-World | 0.2560 | 0.2406 |
| | | Synthetic | 0.5760 | 0.6061 |
| | | Joint | 0.7069 | 0.7198 |
| | Joint | Real-World | 0.6773 | 0.6868 |
| | | Synthetic | 0.5635 | 0.5794 |
| | | Joint | 0.7069 | 0.7198 |
| 4:1 | Real-World | Real-World | 0.4723 | 0.4736 |
| | | Synthetic | 0.4724 | 0.4747 |
| | | Joint | 0.3641 | 0.3494 |
| | Synthetic | Real-World | 0.2556 | 0.2403 |
| | | Synthetic | 0.5767 | 0.6071 |
| | | Joint | 0.7069 | 0.7198 |
| | Joint | Real-World | 0.6772 | 0.6862 |
| | | Synthetic | 0.6204 | 0.6303 |
| | | Joint | 0.7069 | 0.7198 |
| 5:1 | Real-World | Real-World | 0.4724 | 0.4738 |
| | | Synthetic | 0.4724 | 0.4746 |
| | | Joint | 0.3641 | 0.3493 |
| | Synthetic | Real-World | 0.2558 | 0.2407 |
| | | Synthetic | 0.8391 | 0.8460 |
| | | Joint | 0.7073 | 0.7202 |
| | Joint | Real-World | 0.6774 | 0.6863 |
| | | Synthetic | 0.6204 | 0.6304 |
| | | Joint | 0.7073 | 0.7202 |

Table 4: Performance of models trained on sequences of length 100 — Best results overall and for real-world test data are highlighted

## C Results: Models Trained on Data Length 200

| Key Option | Train Data | Test Data | Accuracy | *F1* |
|---|---|---|---|---|
| 1:1 | Real-World | Real-World | 0.3597 | 0.3711 |
| | | Synthetic | 0.3560 | 0.3641 |
| | | Joint | 0.2027 | 0.2056 |
| | Synthetic | Real-World | 0.0457 | 0.0546 |
| | | Synthetic | 0.5491 | 0.5493 |
| | | Joint | 0.2933 | 0.2981 |
| | Joint | Real-World | 0.5216 | 0.5336 |
| | | Synthetic | 0.6105 | 0.6191 |
| | | Joint | 0.2933 | 0.2981 |
| 2:1 | Real-World | Real-World | 0.3548 | 0.3619 |
| | | Synthetic | 0.3547 | 0.3609 |
| | | Joint | 0.2007 | 0.2028 |
| | Synthetic | Real-World | 0.0453 | 0.0536 |
| | | Synthetic | 0.5500 | 0.5502 |
| | | Joint | 0.2935 | 0.2984 |
| | Joint | Real-World | 0.5203 | 0.5310 |
| | | Synthetic | 0.6094 | 0.6174 |
| | | Joint | 0.2935 | 0.2984 |
| 3:1 | Real-World | Real-World | 0.3544 | 0.3605 |
| | | Synthetic | 0.3547 | 0.3609 |
| | | Joint | 0.1998 | 0.2016 |
| | Synthetic | Real-World | 0.0449 | 0.0528 |
| | | Synthetic | 0.5495 | 0.5496 |
| | | Joint | 0.2936 | 0.2985 |
| | Joint | Real-World | 0.5198 | 0.5291 |
| | | Synthetic | 0.6087 | 0.6160 |
| | | Joint | 0.2936 | 0.2985 |
| 4:1 | Real-World | Real-World | 0.3547 | 0.3609 |
| | | Synthetic | 0.3548 | 0.3619 |
| | | Joint | 0.1997 | 0.2015 |
| | Synthetic | Real-World | 0.0448 | 0.0528 |
| | | Synthetic | 0.5497 | 0.5499 |
| | | Joint | 0.2938 | 0.2986 |
| | Joint | Real-World | 0.5209 | 0.5306 |
| | | Synthetic | 0.6095 | 0.6168 |
| | | Joint | 0.2938 | 0.2986 |
| 5:1 | Real-World | Real-World | 0.3544 | 0.3605 |
| | | Synthetic | 0.3547 | 0.3609 |
| | | Joint | 0.1998 | 0.2014 |
| | Synthetic | Real-World | 0.0453 | 0.0533 |
| | | Synthetic | 0.7011 | 0.7164 |
| | | Joint | 0.2927 | 0.2975 |
| | Joint | Real-World | 0.5216 | 0.5336 |
| | | Synthetic | 0.6125 | 0.6239 |
| | | Joint | 0.2927 | 0.2975 |

Table 5: Performance of models trained on sequences of length 200 — Best results overall and for real-world test data are highlighted