The Turing Test?

Vividata Hiring Challenge

The challenge

"No, I'm not interested in developing a powerful brain. All I'm after is just a mediocre brain . . . "
— Alan Turing

There is currently significant debate regarding whether modern neural networks have "passed" the Turing Test, in other words whether a system that you can chat with seamlessly should be regarded as intelligent. But long before the imitation game became a pop-cultural benchmark, Turing was already asking a more concrete (and, frankly, more fun) question: can a machine learn to crack a real cipher?

In the 1940s that question had an existential motivation. The enigma code was used by all branches of the German military, used even for top secret messages it was regarded as effectively unbreakable. It gave the Germans significant advantages in U-boat operations, including The Battle of the Atlantic. Building on the work of Polish mathematician Marian Rejewski, Alan Turing led the seemingly impossible task of automating the codebreaking process.

At Bletchley Park, codebreaking required identifying patterns within incomplete and noisy evidence, while keeping pace with an adversary who altered procedures daily. Turing helped formalize this practice into a disciplined scientific methodology.

This competition invites you to revisit that original challenge. Instead of judging a model by how human it sounds, you will judge it by whether it can recover meaning from Enigma ciphertext.

Your goal is to train a neural network that can *break the German Enigma machine.* Given ciphertext produced by an Enigma configuration, your model should predict the plaintext sequence that generated it. Your model must balance performance with efficiency and elegance.

How the Enigma works (and what must be cracked)

Enigma is a substitution cipher implemented as an *electromechanical* device. What makes it challenging is not a single substitution table, but a substitution that changes at every keystroke.

At a high level, each character is transformed by a signal path like this:

- **Plugboard (Steckerbrett):** optional pairwise swaps applied before and after the rotors.
- **Rotor stack:** a sequence of wired rotors that each apply a permutation.
- **Stepping:** after each keypress (and via turnover notches), the rightmost rotor advances, with cascaded stepping that changes the effective cipher over time.
- **Reflector (UKW):** sends the signal back through the rotors, ensuring encryption is reciprocal (the same settings decrypt).

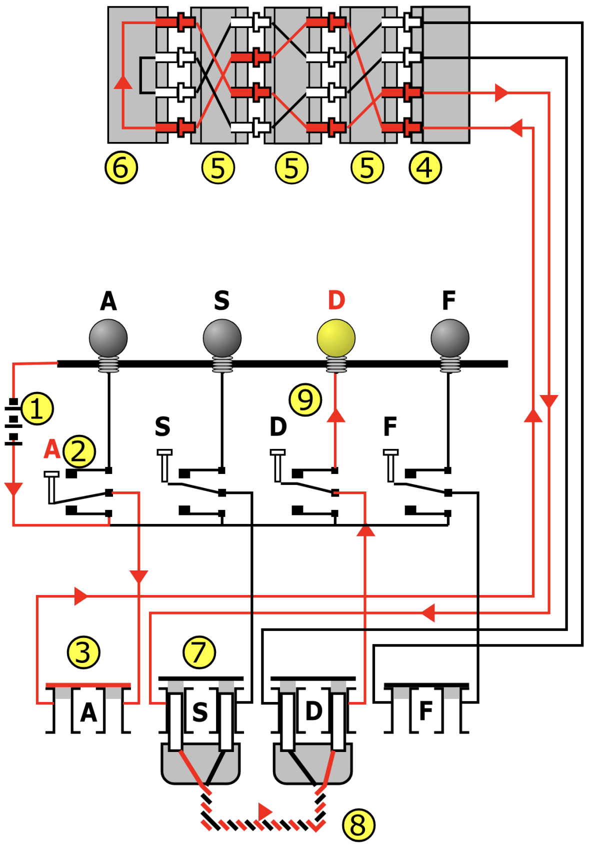

Figure 1: Enigma signal path diagram. Source: By MesserWoland, CC BY-SA 3.0, Wikimedia Commons (curid=1790479).

Which settings are unknown? In classical usage, an operator chooses a daily key (and sometimes a per-message key). To decrypt, you must infer (some or all of) the following:

- **Rotor choice and order** (which rotors, and in what sequence).

- **Ring settings** (Ringstellung): the offset between rotor wiring and the visible alphabet ring.
- **Starting positions** (Grundstellung / message key): the initial rotor letters at the start of encryption.
- **Plugboard wiring** (Steckerverbindungen): which letters are swapped on the plugboard.

For this challenge, we are most interested in breaking the **rotor choice/order and rotor positions** (the moving core of Enigma), while the other elements may remain unmasked for the the model.

For more background, see the Wikipedia article on the Enigma machine.

Which Enigma? For this challenge we will be using the common *Army/Air Force three-rotor* style machine.

We have made a python script simulating enigma to help get you started. It may be found in the 'Useful Materials' at vividata.ai/careers.

Task definition

Input: Ciphertext $c_{1:n}$ (letters A–Z) generated by an Enigma instance.

Output: Predicted plaintext $\hat{p}_{1:n}$.

Unless otherwise stated by the data release, assume:

- Uppercase A–Z only; no spaces or punctuation.
- Training data consists of (ciphertext, English corpus plaintext) pairs produced from randomly sampled Enigma settings.
- Test-time evaluation input is ciphertext only, with the rotor order and settings masked.
- All computation must be performed by the neural network. Therefore no code execution or tool use allowed.

Judging criteria

Submissions will be ranked using a composite score based on:

1. **Accuracy:** character-level and/or sequence-level plaintext recovery on the hidden test set.
2. **Model size:** smaller is better (e.g., parameter count and/or compressed artifact size).
3. **Sample efficiency:** the amount of ciphertext required to decrypt (shorter required samples score higher).

Extensions and ideas

Bonus points are available (but not required) for exploring:

- Transfer from 3-rotor to 4-rotor (Naval) settings.
- Use historic German radio messages solely for training.
- Uncertainty estimation: when does the model *know* it is wrong?

Useful materials

The following resources may be helpful as you train and evaluate your model:

- **English-language evaluation:** in addition to the hidden test set, you can sanity-check your decoder by evaluating it on an English-language corpus (e.g., by measuring character accuracy or sequence accuracy on held-out English text after encrypting/decrypting). A large scale English dataset may be found at `HuggingFaceFW/fineweb`.
- **Enigma reference code:** an attached Python script is provided as a template implementation of the Enigma machine for data generation, debugging, and baselines.

Submission format

You should submit your solution as a technical write up to, please include your experiments and results (what you tried, what worked, what didn't, and why). The format of the write up may take the form of a whitepaper, blogpost or other suitable format.

If possible please also include a runnable inference script, which can convert ciphertext to predicted plaintext. Lastly, please provide a model artifact (weights), along with the model size.

Please send your submissions to **careers@vividata.ai** by the 31st March 2026. The best submissions will be guaranteed an interview with Vividata.

We are working on making this a proper eval – and are taking using this footnote to take the credit for this idea.