The development of a solution to the problem of detoxication can be divided into several stages, each of which was informed by different sources. The solution involves the use of Natural Language Processing (NLP) techniques, specifically machine translation and sequence-to-sequence (Seq2Seq) models. In this report, we will describe the path taken to solve this problem, using a combination of different sources.

The first source, [Paperspace Blog](https://blog.paperspace.com/nlp-machine-translation-with-keras/), provided a comprehensive introduction to machine translation using Keras, a popular deep learning library. This source was instrumental in understanding how to implement a Seq2Seq model for machine translation tasks.

The second source, another [Paperspace Blog](https://blog.paperspace.com/introduction-to-seq2seq-models/), provided a detailed explanation of Seq2Seq models. It explained how these models work, their architecture, and how they are used in tasks like machine translation.

The third source, [Habr](https://habr.com/ru/articles/762140/), provides a Russian article on how to use Keras to build a Seq2Seq model for text generation. This source is similar to the second source, but it provides additional insights into how to use Keras for text generation tasks. These insights are useful for our solution, as they help us understand how to generate detoxified text.

The fourth source, another [Habr article](https://habr.com/ru/articles/581932/), was used to understand how to improve the performance of Seq2Seq models. This source provided techniques to improve the performance of Seq2Seq models, such as using attention mechanisms and bidirectional LSTMs. This source is similar to the first and second sources, but it focuses on machine translation instead of text generation. These concepts are applicable to our solution, as they help us understand how to transform toxic words into non-toxic words.

The final source, a [Towards Data Science article](https://towardsdatascience.com/training-t5-for-paraphrase-generation-ab3b5be151a2), was used to understand how to use the T5 model for paraphrase generation. This source provided a detailed guide on how to train the T5 model and how to use it for paraphrase generation.

Based on the information from these sources, the solution was developed in three stages:

1. **Data Preprocessing**: The first stage involved preprocessing the data. This included tokenizing the sentences, adding start-of-sequence (SOS) and end-of-sequence (EOS) tokens, and converting the sentences into sequences of integers.

2. **Model Training**: The second stage involved training the Baseline dictionary model, Seq2Seq model and T5 Transformer. This stage included creating the models architecture, compiling the model, and training the models using the preprocessed data.

3. **Model Evaluation**: The final stage involved evaluating the model. This included predicting sequences using the trained model and calculating the BLEU score to evaluate the performance of the model.

In conclusion, we would like to specify that the Baseline dictionary based model is too primitive to use in the text detoxification problem because it doesn’t take context of the sentence into account and has only one form of each toxic word. The solution to the problem involves building a Seq2Seq model using Keras and a T5 model. The Seq2Seq model is used to transform input text (which may contain toxic words) into output text (which does not contain toxic words). The solution also involved the use of three different Seq2Seq models: a simple Seq2Seq model, a Seq2Seq model with stacked LSTMs, and a Seq2Seq model with pre-trained word embeddings. These models were used to demonstrate how different model architectures can affect the performance of the model. The T5 model is used to understand the context of the text and generate non-toxic words. The solution involves preprocessing the text data and training the models on this data. The solution is a combination of the concepts introduced in the five sources, with each source contributing to a different aspect of the solution.