1. Source Code Generation from User Intention by Using Recurrent Neural Networks

<https://www.linkedin.com/pulse/source-code-generation-from-user-intention-using-recurrent-kelvin/>

1. Code generation by a generalized neural network: General principles and elementary examples

<https://www.sciencedirect.com/science/article/pii/074373158990066X>

1. Code or (not Code) – Separating Formal and Natural Language in CS Education

<http://www.macs.hw.ac.uk/~greg/publications/ccdm.WIPSCE14.pdf>

1. Studying the Difference Between Natural and Programming Language Corpora

<https://arxiv.org/pdf/1806.02437.pdf>

1. Generating Text with Recurrent Neural Networks

<https://www.cs.utoronto.ca/~ilya/pubs/2011/LANG-RNN.pdf>

1. Generating Sequences With Recurrent Neural Networks

<https://arxiv.org/pdf/1308.0850.pdf>

1. Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization

<https://ieeexplore.ieee.org/abstract/document/1683770>

1. A Language for Writing Code Generators

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.446.521&rep=rep1&type=pdf>

1. Natural Language Processing based Automatic Multilingual Code Generation

<https://pdfs.semanticscholar.org/fee0/d42fb08ef1618dadd83734bea38e7b213f25.pdf>

1. A Neural Network Approach to Context-Sensitive Generation of Conversational Responses

<https://arxiv.org/pdf/1506.06714.pdf>

1. The Unreasonable Effectiveness of Recurrent Neural Networks: Code generation part

<https://karpathy.github.io/2015/05/21/rnn-effectiveness/>

If we can get syntax checking down, we are successful.

1. Gherkin reference

<https://docs.cucumber.io/gherkin/reference/>

1. Code graph

<https://visualstudiomagazine.com/articles/2013/04/25/use-code-maps-to-understand-code-relationships.aspx>

1. Map dependencies with code maps

<https://docs.microsoft.com/en-us/visualstudio/modeling/map-dependencies-across-your-solutions?view=vs-2017>

1. JDave

<http://jdave.org/documentation.html>

1. Concordian

<https://concordion.org/coding/java/markdown/>

1. JBehave

<https://jbehave.org/reference/stable/>

1. Serenity BDD

<http://www.thucydides.info/#/documentation>

1. SpecFlow

<https://specflow.org/docs/>

1. A Comparative Case Study on the Impact of Test-Driven Development on Program Design and Test Coverage

<https://arxiv.org/ftp/arxiv/papers/1711/1711.05082.pdf>

1. An Initial Investigation of Test Driven Development in Industry

<https://collaboration.csc.ncsu.edu/laurie/Papers/TDDpaperv8.pdf>

1. A Literature Review of Behavior Driven Development using Grounded Theory

<https://pdfs.semanticscholar.org/4f03/ec0675d08cfd1ecdbaac3361a29d756ce656.pdf>

1. Code Maps from Visual Studio

<https://docs.microsoft.com/en-us/visualstudio/modeling/map-dependencies-across-your-solutions?view=vs-2017>

1. Architexa

<http://www.architexa.com/features/layered-diagrams>