Agenda

- Homework: reviews of one another’s proposals
- How to present a research paper
- How to present the mid-point check projects
- How to pivot

http://34.204.188.58/WSDM/compare_and_contrast/254564747.html

- **tl;dr:** This work proposes a cross-architecture training procedure with a margin focused loss (Margin-MSE),
  - that adapts knowledge distillation to the varying score output distributions of different BERT and non-BERT ranking architectures,
  - and shows that across evaluated architectures it significantly improves their effectiveness without compromising their efficiency.

- **Chatbot Summary**:
  - As a reviewer, you would want a concise summary of "Improving Efficient Neural Ranking Models with Cross-Architecture Knowledge Distillation."
  - This paper proposes a cross-architecture training method using Margin-MSE loss, adapting knowledge distillation to various BERT and non-BERT ranking architectures.
  - The study demonstrates that this approach significantly enhances the effectiveness of different architectures without sacrificing efficiency. The focus is on reducing latency for efficient ranking models, making them more suitable for real-world deployment.

www.vecml.com

```
from vecml import vecml
import numpy as np
import argparse

parser = argparse.ArgumentParser()
parser.add_argument("--chat_name", help="string", required=True)
parser.add_argument("--files", help="comma separated filenames", default=None)
parser.add_argument("--prompt", help="prompt", required=True)
parser.add_argument("--apikey", help="API key from VecML", required=True)
args = parser.parse_args()

if __name__ == '__main__':
    vecml.init(args.apikey,"us-west")

    if not args.files is None:
        vecml.create_chat(args.chat_name, args.files.split(','))
    print(vecml.chat(args.chat_name, args.prompt))
```