```
from transformers import AutoTokenizer, AutoModelForCausalLM
import torch
import random

# Load ProtGPT2 or equivalent model
tokenizer = AutoTokenizer.from_pretrained("nferruz/ProtGPT2")
model = AutoModelForCausalLM.from_pretrained("nferruz/ProtGPT2")

def generate_binders(fusion_context, strategy='low_shot', num_candidates=10):
    seed_sequence = fusion_context['embedding_vector'][:10]
    seed = ''.join([chr(int(65 + abs(int(x * 10)) % 20)) for x in seed_sequence])
    input_ids = tokenizer.encode(seed, return_tensors="pt")

    outputs = model.generate(
        input_ids,
        do_sample=True,
        top_k=950,
        top_p=0.96,
        temperature=1.0,
        max_length=200,
        num_return_sequences=num_candidates
    )

    binders = []
    for output in outputs:
```

```
    sequence = tokenizer.decode(output, skip_special_tokens=True)
    sequence = ''.join([aa for aa in sequence if aa in "ACDEFGHIKLMNPQRSTVWY"])
    if len(sequence) > 30:
        binders.append(sequence)

return {"generated_binders": binders}
```