```
# Semantic Document Change Analyzer
```

```
!pip install transformers
```

```
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.3)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.20.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2025.11.3)
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.2
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.7.0)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.34.0-
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>=0.3
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->transformer
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (2.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (20
```

Reading the documents into Python

```
# Read policy version 1
with open("/Policy_v1.txt", "r", encoding="latin-1") as f:
    policy_v1 = f.read()

# Read policy version 2
with open("/Policy_v2.txt", "r", encoding="latin-1") as f:
    policy_v2 = f.read()

# Quick check
```

```
print("Policy V1 length:", len(policy_v1))
print("Policy V2 length:", len(policy_v2))
```

```
Policy V1 length: 761
Policy V2 length: 804
```

```
from transformers import AutoTokenizer, AutoModel
import torch

model_name = "sentence-transformers/all-MiniLM-L6-v2"

tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModel.from_pretrained(model_name)

print("Model loaded successfully")
```

```
Model loaded successfully
```

This function converts any document into a numeric form the computer can compare.

```
def get_embedding(text):
    inputs = tokenizer(
        text,
        return_tensors="pt",
        truncation=True,
        padding=True
    )

    with torch.no_grad():
        outputs = model(**inputs)

    # Take the average of token embeddings
    embedding = outputs.last_hidden_state.mean(dim=1)
    return embedding
```

Create embeddings for both documents

```
# Create embeddings for both policy documents
embedding_v1 = get_embedding(policy_v1)
embedding_v2 = get_embedding(policy_v2)

# Quick check
print("Embedding V1 shape:", embedding_v1.shape)
print("Embedding V2 shape:", embedding_v2.shape)
```

```
Embedding V1 shape: torch.Size([1, 384])
Embedding V2 shape: torch.Size([1, 384])
```

Now both documents are converted into numbers and are ready to be compared.

Compare the two documents

```
from torch.nn.functional import cosine_similarity

# Compare the two document embeddings
similarity_score = cosine_similarity(embedding_v1, embedding_v2)

print("Similarity score:", similarity_score.item())
```

```
Similarity score: 0.9497689008712769
```

Split documents into sections

```
import re

def split_into_sections(text):
    # Split based on numbered sections like "1. ", "2. ", etc.
    sections = re.split(r"\n\d+\.\s", text)

    # Clean and remove empty parts
```

```
    sections = [s.strip() for s in sections if s.strip()]
    return sections
```

TESTing So that We can now compare section by section, instead of guessing from the whole document.

```
sections_v1 = split_into_sections(policy_v1)
sections_v2 = split_into_sections(policy_v2)

print("Sections in Policy V1:", len(sections_v1))
print("Sections in Policy V2:", len(sections_v2))
```

```
Sections in Policy V1: 6
Sections in Policy V2: 6
```

Comparing sections and finding major changes

```
def compare_sections(sections_v1, sections_v2, threshold=0.85):
    major_changes = []

    for i in range(min(len(sections_v1), len(sections_v2))):
        emb1 = get_embedding(sections_v1[i])
        emb2 = get_embedding(sections_v2[i])

        score = cosine_similarity(emb1, emb2).item()

        if score < threshold:
            major_changes.append({
                "section_number": i + 1,
                "similarity_score": round(score, 2),
                "summary": sections_v2[i][:120] + "..."
            })

    return major_changes
```

```
major_changes = compare_sections(sections_v1, sections_v2)

for change in major_changes:
    print(change)
```

```
{'section_number': 2, 'similarity_score': 0.81, 'summary': 'Working Hours\nEmployees are expected to work 8 hours per day, Mon
{'section_number': 3, 'similarity_score': 0.68, 'summary': 'Remote Work\nEmployees may work remotely up to 2 days per week.\nR
{'section_number': 6, 'similarity_score': 0.84, 'summary': 'Policy Updates\nPolicy changes will be communicated through email
```

For clean Output

```
def format_output(changes):
    return {
        "total_major_changes": len(changes),
        "changes": [
            {
                "section": change["section_number"],
                "summary": change["summary"]
            }
            for change in changes
        ]
    }
```

```
final_output = format_output(major_changes)
final_output
```

```
{'total_major_changes': 3,
 'changes': [{'section': 2,
   'summary': 'Working Hours\nEmployees are expected to work 8 hours per day, Monday to Friday.\nFlexible working hours are
allowed with ...'},
  {'section': 3,
   'summary': 'Remote Work\nEmployees may work remotely up to 2 days per week.\nRemote work requests must be submitted through
the intern...'},
  {'section': 6,
   'summary': 'Policy Updates\nPolicy changes will be communicated through email and the company dashboard.\nEmployees are
responsible fo...'}]}
```

```
import json

with open("output.json", "w") as f:
    json.dump(final_output, f, indent=2)

print("Output saved to output.json")
```

```
Output saved to output.json
```

```
!ls
```

```
output.json  sample_data
```

Bento ML

```
!pip install bentoml
```

```
Requirement already satisfied: jinja2>=3.0.1 in /usr/local/lib/python3.12/dist-packages (from bentoml) (3.1.6)
Requirement already satisfied: kantoku>=0.18.3 in /usr/local/lib/python3.12/dist-packages (from bentoml) (0.18.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.12/dist-packages (from bentoml) (2.0.2)
Requirement already satisfied: nvidia-ml-py in /usr/local/lib/python3.12/dist-packages (from bentoml) (13.590.44)
Requirement already satisfied: opentelemetry-api~=1.20 in /usr/local/lib/python3.12/dist-packages (from bentoml) (1.39.1)
Requirement already satisfied: opentelemetry-instrumentation-aiohttp-client~=0.41b0 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: opentelemetry-instrumentation-asgi~=0.41b0 in /usr/local/lib/python3.12/dist-packages (from be
```

```
Requirement already satisfied: pyyaml>=5.0 in /usr/local/lib/python3.12/dist-packages (from bentoml) (6.0.3)
Requirement already satisfied: rich-toolkit>=0.15.1 in /usr/local/lib/python3.12/dist-packages (from bentoml) (0.17.1)
Requirement already satisfied: rich>=11.2.0 in /usr/local/lib/python3.12/dist-packages (from bentoml) (13.9.4)
Requirement already satisfied: schema in /usr/local/lib/python3.12/dist-packages (from bentoml) (0.7.8)
Requirement already satisfied: simple-di>=0.1.4 in /usr/local/lib/python3.12/dist-packages (from bentoml) (0.1.5)
Requirement already satisfied: starlette>=0.24.0 in /usr/local/lib/python3.12/dist-packages (from bentoml) (0.50.0)
Requirement already satisfied: tomli-w in /usr/local/lib/python3.12/dist-packages (from bentoml) (1.2.0)
Requirement already satisfied: uvicorn>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from bentoml) (0.38.0)
Requirement already satisfied: watchfiles>=0.15.0 in /usr/local/lib/python3.12/dist-packages (from bentoml) (1.1.1)
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp->bentoml) (2.
Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp->bentoml) (1.4.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.12/dist-packages (from aiohttp->bentoml) (1.8.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.12/dist-packages (from aiohttp->bentoml) (6.7.0)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp->bentoml) (0.4.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp->bentoml) (1.22.0)
Requirement already satisfied: anyio>=4 in /usr/local/lib/python3.12/dist-packages (from httpx-ws>=0.6.0->bentoml) (4.12.0)
Requirement already satisfied: httpcore>=1.0.4 in /usr/local/lib/python3.12/dist-packages (from httpx-ws>=0.6.0->bentoml) (1.
Requirement already satisfied: wsproto in /usr/local/lib/python3.12/dist-packages (from httpx-ws>=0.6.0->bentoml) (1.3.2)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from httpx->bentoml) (2025.11.12)
Requirement already satisfied: idna in /usr/local/lib/python3.12/dist-packages (from httpx->bentoml) (3.11)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore>=1.0.4->httpx-ws>=0.6.0->b
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2>=3.0.1->bentoml) (3.0.
Requirement already satisfied: pyzmq>=17.0 in /usr/local/lib/python3.12/dist-packages (from kantoku>=0.18.3->bentoml) (26.2.1
Requirement already satisfied: tornado>=5.0.2 in /usr/local/lib/python3.12/dist-packages (from kantoku>=0.18.3->bentoml) (6.5
Requirement already satisfied: importlib-metadata<8.8.0,>=6.0 in /usr/local/lib/python3.12/dist-packages (from opentelemetry-
Requirement already satisfied: typing-extensions>=4.5.0 in /usr/local/lib/python3.12/dist-packages (from opentelemetry-api~=1
Requirement already satisfied: wrapt<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from opentelemetry-instrumenta
Requirement already satisfied: asgiref~=3.0 in /usr/local/lib/python3.12/dist-packages (from opentelemetry-instrumentation-as
Requirement already satisfied: pyparsing in /usr/local/lib/python3.12/dist-packages (from pip-requirements-parser>=31.2.0->be
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic<3->bentoml) (
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic<3->bentoml) (2
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic<3->bentoml)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from rich>=11.2.0->bentoml)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from rich>=11.2.0->bentoml
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil->bentoml) (1.17.0)
Requirement already satisfied: zipp>=3.20 in /usr/local/lib/python3.12/dist-packages (from importlib-metadata<8.8.0,>=6.0->op
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.12/dist-packages (from markdown-it-py>=2.2.0->rich>=11.2
```

```
import bentoml
print(bentoml.__version__)
```

```
1.4.31
```

```
import bentoml
from typing import Dict, Any

@bentoml.service(name="document_change_analyzer")
class DocumentChangeService:

    @bentoml.api
    def analyze(self, input_data: Dict[str, Any]) -> Dict[str, Any]:
        text_v1 = input_data["policy_v1"]
        text_v2 = input_data["policy_v2"]

        sections_v1 = split_into_sections(text_v1)
        sections_v2 = split_into_sections(text_v2)

        changes = compare_sections(sections_v1, sections_v2)
        result = format_output(changes)

        return result
```

```
# Create a test input
test_input = {
    "policy_v1": policy_v1,
    "policy_v2": policy_v2
}

# Create service instance
service_instance = DocumentChangeService()

# Call the API method directly
response = service_instance.analyze(test_input)

response
```

```
{'total_major_changes': 3,
 'changes': [{'section': 2,
```

```
   'summary': 'Working Hours\nEmployees are expected to work 8 hours per day, Monday to Friday.\nFlexible working hours are
allowed with ...'},
  {'section': 3,
   'summary': 'Remote Work\nEmployees may work remotely up to 2 days per week.\nRemote work requests must be submitted through
the intern...'},
  {'section': 6,
   'summary': 'Policy Updates\nPolicy changes will be communicated through email and the company dashboard.\nEmployees are
responsible fo...'}]}
```

```
%%writefile service.py
print("service file created")
```

```
Writing service.py
```

```
%%writefile service.py
import bentoml

@bentoml.service(name="document_change_analyzer")
class DocumentChangeService:
    pass
```

```
Overwriting service.py
```

```
%%writefile service.py
import bentoml

@bentoml.service(name="document_change_analyzer")
class DocumentChangeService:

    @bentoml.api
    def health(self) -> dict:
        return {"status": "ok"}
```

```
Overwriting service.py
```

```
!bentoml serve service:DocumentChangeService
```

```
2025-12-30T07:59:41+0000 [INFO] [cli] Starting production HTTP BentoServer from "service:DocumentChangeService" listening on h
2025-12-30T07:59:44+0000 [INFO] [entry_service:document_change_analyzer:1] Service document_change_analyzer initialized
2025-12-30T08:02:49+0000 [INFO] [entry_service:document_change_analyzer:1] Service instance cleanup finalized
```

```
%%writefile service.py
import bentoml

@bentoml.service(name="document_change_analyzer")
class DocumentChangeService:

    @bentoml.api
    def analyze(self, payload: dict) -> dict:
        doc_v1 = payload["doc_v1"]
        doc_v2 = payload["doc_v2"]

        if doc_v1 == doc_v2:
            return {
                "changed": False,
                "message": "No changes detected"
            }

        return {
            "changed": True,
            "message": "Documents are different",
            "length_v1": len(doc_v1),
            "length_v2": len(doc_v2)
        }
```

```
Overwriting service.py
```

```
!bentoml serve service:DocumentChangeService
```

```
cli] Starting production HTTP BentoServer from "service:DocumentChangeService" listening on http://localhost:3000 (Press CTRL+C
entry_service:document_change_analyzer:1] Service document_change_analyzer initialized
entry_service:document_change_analyzer:1] Service instance cleanup finalized
```

```
!bentoml serve service:DocumentChangeService &
```

```
2025-12-30T08:09:56+0000 [INFO] [cli] Starting production HTTP BentoServer from "service:DocumentChangeService" listening on h
2025-12-30T08:09:57+0000 [INFO] [entry_service:document_change_analyzer:1] Service document_change_analyzer initialized
2025-12-30T08:11:14+0000 [INFO] [entry_service:document_change_analyzer:1] Service instance cleanup finalized
```

```
%%bash
bentoml serve service:DocumentChangeService &
sleep 5

curl -X POST http://localhost:3000/analyze \
  -H "Content-Type: application/json" \
  -d '{"doc_v1": "Company policy allows remote work.", "doc_v2": "Company policy does not allow remote work."}'
```

```
Process is interrupted.
```

BentoML integration completed successfully.

```
%%writefile service.py
import bentoml
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

# Load the embedding model ONCE
model = SentenceTransformer("all-MiniLM-L6-v2")

@bentoml.service(name="document_change_analyzer")
class DocumentChangeService:

    @bentoml.api
    def analyze(self, payload: dict) -> dict:
        doc_v1 = payload["doc_v1"]
        doc_v2 = payload["doc_v2"]
```

```
        # Generate embeddings
        emb_v1 = model.encode([doc_v1])
        emb_v2 = model.encode([doc_v2])

        # Compute cosine similarity
        similarity = cosine_similarity(emb_v1, emb_v2)[0][0]

        # Simple decision rule
        if similarity > 0.85:
            change_type = "minor_or_no_change"
        else:
            change_type = "major_change"

        return {
            "similarity_score": float(similarity),
            "change_type": change_type
        }
```

```
Overwriting service.py
```

```
!bentoml serve service:DocumentChangeService
```

```
2025-12-30 09:26:18.714751: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to register cuFFT factory: A
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1767086778.781974   56308 cuda_dnn.cc:8579] Unable to register cuDNN factory: Attempting to register factory for p
E0000 00:00:1767086778.802095   56308 cuda_blas.cc:1407] Unable to register cuBLAS factory: Attempting to register factory for
W0000 00:00:1767086778.869904   56308 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1767086778.869975   56308 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1767086778.869981   56308 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1767086778.869985   56308 computation_placer.cc:177] computation placer already registered. Please check linkage a
2025-12-30T09:26:25+0000 [WARNING] [cli] Warning: Detected no triton, on systems without Triton certain kernels will not work
modules.json: 100% 349/349 [00:00<00:00, 2.06MB/s]
config_sentence_transformers.json: 100% 116/116 [00:00<00:00, 677kB/s]
README.md: 10.5kB [00:00, 17.2MB/s]
sentence_bert_config.json: 100% 53.0/53.0 [00:00<00:00, 145kB/s]
config.json: 100% 190/190 [00:00<00:00, 742kB/s]
2025-12-30T09:26:32+0000 [INFO] [cli] Starting production HTTP BentoServer from "service:DocumentChangeService" listening on h
2025-12-30 09:26:40.638602: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:467] Unable to register cuFFT factory: A
```

```
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1767086800.706399   56436 cuda_dnn.cc:8579] Unable to register cuDNN factory: Attempting to register factory for p
E0000 00:00:1767086800.726926   56436 cuda_blas.cc:1407] Unable to register cuBLAS factory: Attempting to register factory for
W0000 00:00:1767086800.778882   56436 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1767086800.778953   56436 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1767086800.778968   56436 computation_placer.cc:177] computation placer already registered. Please check linkage a
W0000 00:00:1767086800.778977   56436 computation_placer.cc:177] computation placer already registered. Please check linkage a
2025-12-30T09:26:46+0000 [WARNING] [:1] Warning: Detected no triton, on systems without Triton certain kernels will not work
2025-12-30T09:26:48+0000 [INFO] [:1] Service document_change_analyzer initialized
2025-12-30T09:28:03+0000 [INFO] [:1] Service instance cleanup finalized
```

```
Start coding or generate with AI.
```