实训2：文本词频分析与可视化

· ***目标***：掌握文本处理和Matplotlib。

· ***任务***：读取文本文件，清洗数据，统计词频，绘制柱状图。

· ***详细要求***：

o 读取UTF-8编码文本文件（约1000字），如新闻或小说片段。

o 使用正则表达式移除标点符号，文本转为小写。

o 加载停用词表（stopwords.txt，包含“的”“是”等），过滤停用词。

o 统计词频，输出前10高频词及其出现次数。

o 使用Matplotlib绘制柱状图，横轴为单词，纵轴为频率，添加标题、标签，旋转横轴标签45度。

o 保存词频结果到word_freq.txt。

o 代码需包含错误处理（如文件编码错误、停用词文件缺失）。

o 添加参数配置（如自定义前N高频词）。

· ***技能***：正则表达式、Counter、Matplotlib绘图、文件操作。

```
# -*- coding: utf-8 -*-
import re
import argparse
from collections import Counter

def load_stopwords(file_path):
    try:
        with open(file_path, 'r', encoding='utf-8') as f:
            return set(line.strip() for line in f if line.strip())
    except FileNotFoundError:
        print(f"错误：未找到停用词文件 {file_path}")
        return set()

def clean_text(text):
    # 去除标点符号，只保留字母和汉字
    text = re.sub(r'[^\w\s]', '', text)
    return text.lower()

def process_text_file(text_file, stopwords, topn):
    try:
        with open(text_file, 'r', encoding='utf-8') as f:
            text = f.read()
    except UnicodeDecodeError:
        print(f"错误：无法读取文件 '{text_file}'，请确认编码为 UTF-8。")
        return

    cleaned_text = clean_text(text)
    words = cleaned_text.split()
    filtered_words = [word for word in words if word not in stopwords]
    word_counts = Counter(filtered_words)
    top_words = word_counts.most_common(topn)

    # 写入文件
```

```
    try:
        with open('word_freq.txt', 'w', encoding='utf-8') as f:
            for word, count in top_words:
                f.write(f"{word}\t{count}\n")
        print(f"已保存词频结果到 word_freq.txt")
    except Exception as e:
        print(f"写入文件时出错：{e}")

def main():
    parser = argparse.ArgumentParser(description="文本词频统计（不含可视化）")
    parser.add_argument('--text', type=str, default='sample.txt', help='输入的文本文件路径')
    parser.add_argument('--stopwords', type=str, default='stopwords.txt', help='停用词文件路径')
    parser.add_argument('--topn', type=int, default=10, help='输出前N个高频词')

    args = parser.parse_args()

    stopwords = load_stopwords(args.stopwords)
    process_text_file(args.text, stopwords, args.topn)

if __name__ == '__main__':
    main()
```