توضیح مختصر کد از ابتدا تا به اینجا:

در ابتدا، فایلی با نام "IR_data_news_12k.json" با فرمت utf8 می‌خواند و اطلاعات موجود در آن را با استفاده از تابع json.load()فراخوانی می‌کند. اسم فایل و مسیرش به صورت نسبی (در پوشه بالاتر) در تابع open() ورودی داده شده است.

در تابعnormalize() ، متن هر سند ابتدا با استفاده از کتابخانه‌ی nltk و توابع normalizer()وword_tokenize() ، پاکسازی و توکن‌بندی می‌شود. سپس با استفاده از تابع stemmer() اعداد و انواع مختلف "واژه" در متن به ریشه‌یابی می‌شوند. کلمات پایه‌ای که در لیست stopWordsList موجود هستند حذف می‌شوند. در نهایت، تابع index() برای بررسی وجود هر کلمه در متن سند استفاده می‌شود و در صورت وجود، واژه به لیست پوزیشنال اضافه می‌شود.

در تابعnotIn() ، اگر یک واژه در لیست positionalIndex موجود باشد، تمامی اسناد مربوطه را از لیست داکیومنت‌های مورد علاقه (docsRanks)حذف می‌کند.

در تابعphrasal() ، عبارت ورودی تابع (words) به بخش‌هایی تقسیم می‌شود و در لیستی به نام phrases قرار می‌گیرد.

در تابعpositionCheck() ، بررسی می‌شود که آیا کلمه مربوط به جمله اول عبارت چندکلمه‌ای در این شناسه سند وجود دارد یا خیر و در صورت وجود واژه، آیا کلمات بعدی عبارت چندکلمه‌ای در جایگاه مناسب (یعنی پس از کلمه قبلی) قرار دارند؟ در صورتی که تمامی کلمات عبارت چندکلمه‌ای در جایگاه مناسب قرار داشته باشند، مجموعه سند مربوطه به لیست docsRanks اضافه می‌شود.

در تابعphraseRank() ، خطوط کد جدیدی برای ارزیابی اسناد بر اساس عبارت چندکلمه‌ای اضافه می‌شود. برای این کار، ابتدا یک کپی از docsContent و positionalIndex تهیه می‌شود. سپس، تمامی واژه‌های یافت شده در عبارت چندکلمه‌ای بررسی می‌شود. در صورتی که هر کلمه در عبارت چندکلمه‌ای وجود نداشت، تابع پایان می‌یابد. در غیر این صورت، واژه‌هایی که در لیست positionalIndexموجود نیستند، حذف می‌شوند و سپس، تمامی سندهایی که از عبارت چندکلمه‌ای شامل همه کلمات آن نیستند، بطور کامل از متن حذف می‌شوند. سپس، تمامی اسنادی که در بخش ۱ لیست positionalIndex برای عبارت چندکلمه‌ای وجود دارند، به لیست docsRanks اضافه می‌شوند.

در تابعprintDocs() ، لیست docsRanks برای چاپ کردن محتوای مربوط به مراجع روی صفحه استفاده می‌شود.