بازیابی پیشرفته‌ی اطلاعات نیم‌سال اول ۱۳۹۹ـ۱۳۹۸

گزارش فاز اول پروژه

مدرس: دکتر بیگی اعضای گروه: احسان سلطان‌آقایی، وحید بالازاده مرشت

مشارکت وحید بالازاده

درصد مشارکت: ۵۰ درصد.

کارهای انجام‌گرفته:

- بخش ۲: کد این بخش در فایل indexer.py و در کلاس Indexer قابل مشاهده است. در این بخش با استفاده از متودهای add_doc و del_doc می‌توان سندی را به نمایه اضافه و یا حذف کرد. از فیلدهای این کلاس در قسمت جستجو و بازیابی نیز استفاده می‌شود.
- بخش ۵: کد این بخش در فایل search.py و در کلاس Searcher قرار دارد. دو متد search و search_prox به ترتیب برای جستجوی عادی و جستجوی proximity استفاده می‌شوند. در هر دو متود روش lnc.ltc به کار رفته‌است.
- تست کلاس‌های Searcher و Indexer نیز در فایل test_search.py آمده‌است.
- کد مربوط به بخش کنسول که در فایل main.py قرار دارد.

مشارکت احسان سلطان‌آقایی

درصد مشارکت: ۵۰ درصد.

کارهای انجام‌گرفته:

- بخش ۱: کد این بخش در فولدر preprocess موجود است. در این قسمت داده‌های فارسی به فرمت xml خوانده می‌شوند. از مجموعه داده‌های فارسی قسمت عنوان و متن صفحه‌های ویکی‌پدیا تحت یک متن به پیش‌پردازشگر داده می‌شود. از مجموعه داده‌های انگلیسی نیز عنوان و متن اخبار تحت یک متن به پیش‌پردازشگر داده می‌شود. پیش‌پردازشگر انگلیسی از کتابخانه NLTK و پیش‌پردازشگر فارسی از کتابخانه hazm استفاده می‌کند. هم‌چنین درصد معقولی از کلمات پرتکرار با پردازش متن و به کمک نمایش آن حذف می‌شوند.
- بخش ۳: کد این بخش در فولدر compression قرار دارد. نمایه ساخته‌شده در بخش ۲ را دریافت می‌کند و به دو روش variable byte و gamma code فشرده‌سازی می‌شود. نتیجه میزان حافظه اشغال شده به این صورت است که ذخیره سازی به صورت عادی ۱۴ مگابایت، ذخیره‌سازی به روش variable byte حدود ۶ مگابایت و ذخیره‌سازی به روش gomma code حدود ۷ مگابایت فضا اشغال می‌کند.
- بخش ۴: کد این بخش در فولدر edit query موجود است. یک پرسمان دریافت می‌کند. ابتدا تشخیص می‌دهد که فارسی است یا انگلیسی، سپس پیش‌پردازش متناسب را روی آن انجام می‌دهد. سپس کلمات پرسمان را به ترتیب با کلمات نمایه به روش bigram و با معیار jaccard مقایسه می‌کند. در نهایت نزدیک ترین کلمه از بین کلمات منتخب به روش ذکر شده را با معیار edit distance جایگزین کلمه پرسمان می‌کند.