در انجام این پروژه از کتاب خانه قدرت مند nltk استفاده شد که با قابلیت تنظیم مدل، توانایی استخراج ویژگی های مهمی از متون را دارد.

در ابتدا document ها در RAM ذخیره می شود و سپس به بلوک هایی با اندازه از پیش تعریف شده تقسیم می شوند. لازم به ذکر است که در پروژه هایی که اندازه کلی document ها چشمگیر است، تمام document ها یکجا به رم انتقال نمی یابند بلکه این کار به ازای هر document که در حال محاسبه آن هستیم، انجام می شود.

برای هر document در هر بلوک، preprocess انجام می شود که در ابتدا document به توکن های متناظرش تبدیل می شود و سپس punctuation ها حذف می شود (علامت ها و ...) و در نهایت Stemming انجام می شود (توکن ها به حالت root خود می روند.) نتیجه این تابع به ازای هر document مجموعه ای از term های آن document می باشد.

حال به ازای هر term، به روش posting، Inverted index محاسبه می شود.

در این قسمت دو روش کلی برای محاسبت Inverted index داریم:

۱- استفاده از لیست برای ذخیره document هایی که term مذکور را شامل هستند. در این صورت در این لیست ممکن است document های تکراری ببینیم.
۲- استفاده از set برای ذخیره document هایی که term مذکور را شامل هستند. در این صورت دیگر document تکراری نمی بینیم و باید تعداد تکرار term در هر document در ارایه ای دیگر مشخص شود.

این دور روش advantages و disadvantages های خاص خود را دارند. برای مثال اگر یک term به تعداد زیادی در document اورده شود، اگر از روش ۱ استفاده کنیم تعداد زیادی کد یکسان خواهیم داشت اما در روش دوم این مشکل حل می شود.

حال پس از محاسبه inverted index برای بلوک، آن را به وسیله روش gamma gaps، encode می کنیم تا بتوانیم در دیسک (فرضی) بهتر ذخیره کنیم. همانطور که می دانیم روش gamma، به دو حالت قابل پیاده سازی است. یک به این صورت که unary عدد با ۰ نمایش داده شود و دیگری این که با یک نمایش داده شود. در هر صورت فرقی در مسئله نخواهد داشت. هم چنین می توانیم از اعمال bit برای محاسبه آن استفاده کنیم.

سپس بلوک جدید را با حاصل به دست امده از بلوک های قبلی merge می کنیم. (طبیعتا چون از string استفاده شده است، تنها با جمع کردن merge می شود. اما اگر از bit استفاده می کردیم، باید از shift استفاده می کردیم.)

لازم به ذکر است که در این قسمت (merge block) در حالت واقعی (document های چشمگیر)، باید از دیسک نتیجه قبلی خوانده و اضافه می شد.