Creating the Search Request:

You should be able to receive expanded content by setting 2 fields in the request body, `maxSnippetSize,` and `returnLlmContentOverSnippets` .

`maxSnippetSize, int`

Gives a hint to the server on how many characters of expanded content should be returned per result. The server may return less or more.

This field must be set if `returnLlmContentOverSnippets` is true, and it must be an integer value > 0 and <= 10000.

`returnLlmContentOverSnippets, boolean`

If true, the server will return expanded content. If not, the server will return SERP results as expected.

The current retrieval process and formatting of content are what we use internally as LLM content, which we are still modifying for internal usage.

Sample request body (contains all required fields at minimum to get expanded content):

```
None
{
      "query":"mentions",
      "pageSize":10,
      "maxSnippetSize":4000,
      "requestOptions":
      {
            "returnLlmContentOverSnippets":true
      }
}
```

Reading the Search Response:

The LLM content is formatted differently than the non-LLM content you get from a regular `/search` request.

In the response, there is a top-level field `results` . `results` is an array of objects, each representing a search result. Each search result has a top-level field `snippets` which is an array of text content from the search result.

Non-LLM content:

The format of the content is different from LLM content in that the snippets are ordered by score, not in the order they appear in the document, and we include formatting instructions to bold terms in the content that match the query.

`mimeType` : The type of text
`ranges` : The indexes of text that we should apply special formatting to, to indicate a term matching in the query

- The `endIndex` is where the formatting should end
- The `startIndex` is when the formatting should begin
- The type is the kind of formatting we should do on this range.

`snippet` : deprecated
`fullTextList`: For slack/other messaging documents only, the content.
`text` : content
`snippetTextOrdering`: The order of where the snippet appears in the original document.

```
None
"results": [
    {
       ... more fields...
       "snippets":[
         {
           "mimeType":"text/plain",
           "ranges":[
             {
               "endIndex":16,"startIndex":8,"type":"BOLD"
             }
           ],
           "snippet":"",
           "snippetTextOrdering":1
           "text":"Testing mentions"
         },
         {
           "mimeType":"text/plain",
           "ranges":[
             {
```

```
            "endIndex":13,"startIndex":5,"type":"BOLD"
          }
        ],
        "snippet":"",
        "snippetTextOrdering":2
        "text":"This mentions user 1"
      }
          ... more snippets ...
    ],
     ... more fields ...
  },
```

LLM content:

We do not bold hits for LLM content so we do not include the `ranges` field.

`mimeType` stays the same.

For slack documents, we still only populate `fullTextList` for content.

We order snippets in order of the document, not by score, so we do not include `snippetTextOrdering`.

None

```
"results": [
    {
       ... more fields...
       "snippets": [
        {
          "mimeType": "text/plain",
          "text": "Testing mentions",
          "snippet": ""
        },
        {
          "mimeType": "text/plain",
          "text": "This mentions user 1"
          "snippet": ""
        }
            ... more snippets ...
```

```
    ],
     ... more fields ...
  },
```

FAQ:

1. Will maxSnippetSize be honored even when returnLlmContentOverSnippets is set to False or is it clamped at 255 characters?
   We will try to hit the maxSnippetSize passed in, it is not clamped at 255 characters. However this is still a “hint” number and we may not meet it exactly due to wanting to respect word boundaries, and also due to the number of matches to the query there are in the text content.

2. Glean team to confirm that no text is "generated" or "manufactured" when using the returnLlmContentOverSnippets flag vs. additional text being returned verbatim from the original document :
   No text is generated, all text returned should be also present in the original doc. We might not return any snippet at all if there is not enough content to return.

3. Is there a cap on the maximum number of snippets returned per document as part of the search response? If yes, how do we paginate over all the matching snippets from a single document?
   Yes, there is a cap at 10,000 characters. We have found that for our LLM purposes that use snippets, we’ve never needed more than 5000 characters for snippets, so we introduced a cap on both character count and doc count as to prevent sending too large of a response back.

4. I see a lot of content not relevant to the query being sent back when using returnLlmContentOverSnippets – how come?
   When we return snippets (`returnLlmContentOverSnippets` = false), snippets are limited to around 255 characters, so almost always, all the content returned is content related to or matching the search query.

When you ask for LLM Content (`returnLlmContentOverSnippets` = true), you can request up to 10,000 characters using `maxSnippetSize`, and we will try to return as many characters as requested.

So if the document does not have any more matches/relevant content to the search query, we will add extra content that surrounds the matches/relevant content to hit `maxSnippetSize`.

i.e. if the query is `"test"` and the document is in the text block below, we would get a snippet of `"I don't like taking tests"`.

But if `returnLlmContentOverSnippets` = true and we have a larger `maxSnippetSize` than the length of `"I don't like taking tests"`, then we would try to expand the content returned to the text above and below `"I don't like taking tests"` and include the text in `"My favorite…"` and `"Today is.."` in the content returned.

```
None
       ...more text...

       My favorite fruit is apples.

       I don't like taking tests.

       Today is Monday.

       ... more text ...
```

The surrounding text we include may or may not be relevant content, and that is why you may see content that is not relevant to the search query.

Tips on Usage:

If you only want relevant content, then you can use snippets directly (set `returnLlmContentOverSnippets` = false).

You can also increase/decrease the `maxSnippetSize`, an increase would introduce more surrounding content that is potentially irrelevant.

For our LLM usage, we use 4000 for `maxSnippetSize`, which includes a good amount of surrounding content. While the surrounding content would not be a semantic match, it can sometimes provide useful context that gives us better quality answers.

Still in development:

We are constantly making improvements to our content retrieval, and have several projects in flight to improve the quality of content retrieved. You can expect that this API will stay up to date with what we currently use in our LLM product!