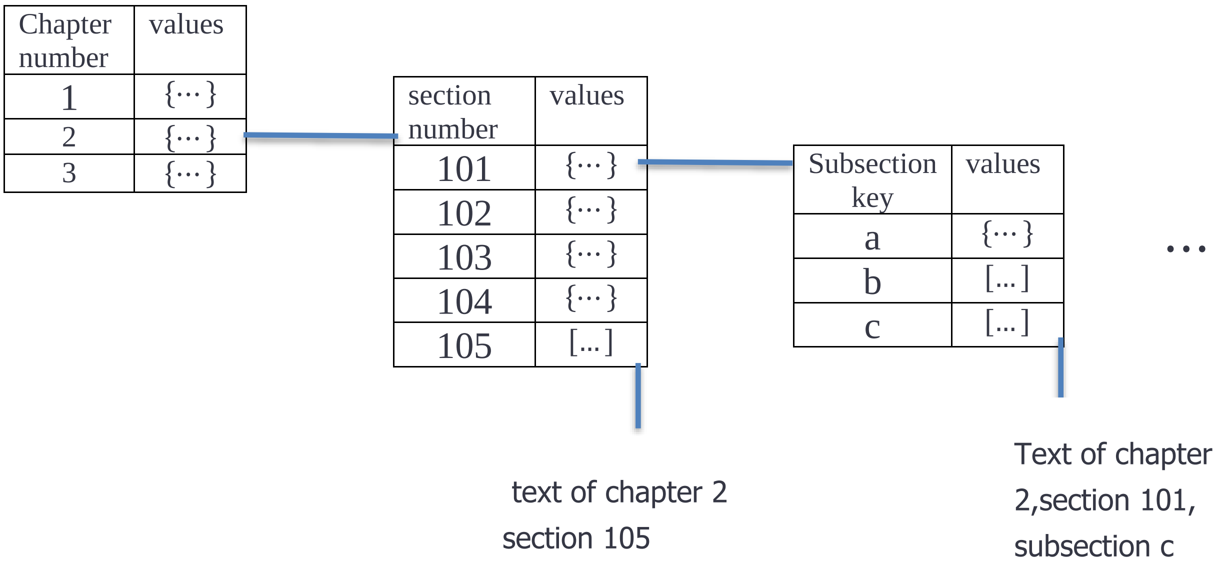
Complete parsing process

The XML file for each title is subdivided into chapters and then sections, but the structure is not even. Often there are optional upper levels, such as subtitles and subchapters, and optional subdivisions, like subsections, paragraphs, subparagraphs, and clauses.

We wrote a script that takes just the text of the regulations and puts it into a dictionary respecting the original structure of the data as closely as possible. This dictionary holds as keys the chapter numbers and as values other dictionaries which in turn hold as keys sections numbers and as values text if there is no further subdivision or another nested dictionary and so on.



Chapter number	values
1	{⋯}
2	{⋯}
3	{⋯}

section number	values
101	{⋯}
102	{⋯}
103	{⋯}
104	{⋯}
105	[...]

Subsection key	values
a	{⋯}
b	[...]
c	[...]

From this structure we extracted a cleaner dictionary we could work with, having as keys strings including each index of each subdivision and as value the text corresponding to that division (e.g. “5.I.12.1202.a” is the key associated to “The term of office of each member of the Merit Systems Protection Board is 7 years.”).

But when starting to run the first demos utilizing the values of this dictionary as our collection of documents, we realized that some subdivisions contained too little information to be useful, thus outputting results apparently not relevant to our queries.

We decided to rework the dictionary once more, creating a final version that holds as keys a string “title.chapter.section” and as value all the text in the section, ignoring any another uplevels or subdivisions present in the data (e.g. the key used in the previous example “5.I.12.1202.a” would just become “52.12.1202” and all the subsections in section 1202 would be included in it’s value)

key	values
1.1.1	[...]
1.1.2	[...]
1.1.3	[...]
1.1.4	[...]
⋮	⋮
1.2.101	[...]
1.2.102	[...]
⋮	⋮