Smart case”：如果查询包含大写字符，它将处理大小写敏感的查询;如果不包含大小写敏感的查询，它将处理大小写不敏感的查询。

例如，一个文档包含asdf、asdF

启用“smart case”后，查询asdf匹配第1行和第2行。查询asdF只匹配第2行。

它也适用于正则表达式。regex查询asd\S匹配第1行和第2行。查询asd[A-Z]只匹配第2行。

引入一个设置：search.smartCase或editor.find.smartCase。这很容易与ripgrep挂钩，但是还需要在编辑器提供结果和未使用ripgrep时实现它。这对于非regex查询很容易。对于regex查询，ripgrep之前尝试判断regex是否匹配任何大写字符。但是这产生了意外的结果，比如\w触发大小写敏感，所以现在它只检查regex文本是否有任何未转义的大写字符(BurntSushi/ripgrep#717)。幸运的是，这很容易匹配。

另外，“区分大小写”的切换按钮可以是一个3模式开关，也可以启用智能大小写。

实现search.smartCase 设置

https://github.com/microsoft/vscode/commit/6c6364d3b9340143da9de2101e702707b628b213

src/vs/platform/search/common/search.ts

+Line98: isSmartCase?: boolean;

+Line163: smartCase: boolean;

src/vs/workbench/parts/search/browser/searchViewlet.ts:

-Line33: import { ISearchProgressItem, ISearchComplete, ISearchQuery, IQueryOptions, ISearchConfiguration } from 'vs/platform/search/common/search';

+Line33: import { ISearchProgressItem, ISearchComplete, ISearchQuery, IQueryOptions, ISearchConfiguration, IPatternInfo } from 'vs/platform/search/common/search';

-Line983: const content = {

+Line983:const content: IPatternInfo = {

-Line988:wordSeparators:this.configurationService.getValue<ISearchConfiguration> ()editor.wordSeparators

+Line988:wordSeparators: this.configurationService.getValue<ISearchConfiguration>().editor.wordSeparators,

+Line989:isSmartCase: this.configurationService.getValue<ISearchConfiguration>().search.smartCase};

src/vs/workbench/parts/search/common/queryBuilder.ts

+Line73: this.resolveSmartCaseToCaseSensitive(contentPattern);

+Line100:

/\*\*

\* Fix the isCaseSensitive flag based on the query and the isSmartCase flag, for search providers that don't support smart case natively.

\*/

private resolveSmartCaseToCaseSensitive(contentPattern: IPatternInfo): void {

if (contentPattern.isSmartCase) {

if (contentPattern.isRegExp) {

// Consider it case sensitive if it contains an unescaped capital letter

if (contentPattern.pattern.match(/([^\\]|^)[A-Z]/)) {

contentPattern.isCaseSensitive = true;

}

} else if (contentPattern.pattern.match(/[A-Z]/)) {

contentPattern.isCaseSensitive = true;

}

}

}

src/vs/workbench/parts/search/electron-browser/search.contribution.ts:

+Line424: },

'search.smartCase': {

'type': 'boolean',

'description': nls.localize('search.smartCase', "Searches case-insensitively if the pattern is all lowercase, otherwise, searches case-sensitively"),

'default': false

src/vs/workbench/parts/search/test/common/queryBuilder.test.ts

+Line605:

suite('smartCase', () => {

test('no flags -> no change', () => {

const query = queryBuilder.text(

{

pattern: 'a'

},

[]);

assert(!query.contentPattern.isCaseSensitive);

});

test('maintains isCaseSensitive when smartCase not set', () => {

const query = queryBuilder.text(

{

pattern: 'a',

isCaseSensitive: true

},

[]);

assert(query.contentPattern.isCaseSensitive);

});

test('maintains isCaseSensitive when smartCase set', () => {

const query = queryBuilder.text(

{

pattern: 'a',

isCaseSensitive: true,

isSmartCase: true

},

[]);

assert(query.contentPattern.isCaseSensitive);

});

test('smartCase determines not case sensitive', () => {

const query = queryBuilder.text(

{

pattern: 'abcd',

isSmartCase: true

},

[]);

assert(!query.contentPattern.isCaseSensitive);

});

test('smartCase determines case sensitive', () => {

const query = queryBuilder.text(

{

pattern: 'abCd',

isSmartCase: true

},

[]);

assert(query.contentPattern.isCaseSensitive);

});

test('smartCase determines not case sensitive (regex)', () => {

const query = queryBuilder.text(

{

pattern: 'ab\\Sd',

isRegExp: true,

isSmartCase: true

},

[]);

assert(!query.contentPattern.isCaseSensitive);

});

test('smartCase determines case sensitive (regex)', () => {

const query = queryBuilder.text(

{

pattern: 'ab[A-Z]d',

isRegExp: true,

isSmartCase: true

},

[]);

assert(query.contentPattern.isCaseSensitive);

});

});

src/vs/workbench/services/search/node/ripgrepTextSearch.ts:

-Line426: args.push(config.contentPattern.isCaseSensitive ? '--case-sensitive' : '--ignore-case');

+Line426: if (config.contentPattern.isSmartCase) {

args.push('--smart-case');

} else {

args.push(config.contentPattern.isCaseSensitive ? '--case-sensitive' : '--ignore-case');

}