LocalAccumulateBy

```java
public <I extends java.io.Serializable> GearsBuilder<I> localAccumulateBy(gears.operations.ExtractorOperation<T> extractor,
gears.operations.AccumulateByOperation<T,I> accumulator)
```

The `localAccumulateBy` function is similar to `accumulateBy`, except it performs the operation locally on each shard without moving data between shards.

On each shard, it iterates through the records in the pipe, groups them based on the provided extractor, and then reduces each group to a single record per group with the accumulator function.

The initial value of the accumulator is null.

**Parameters**

**Type parameters:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>The template type of the returned builder</td>
</tr>
</tbody>
</table>

**Function parameters:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accumulator</td>
<td>AccumulateByOperation&lt;T,I&gt;</td>
<td>A function with logic to update the accumulator value with each record</td>
</tr>
<tr>
<td>extractor</td>
<td>ExtractorOperation</td>
<td>Extracts a specific value from each record</td>
</tr>
</tbody>
</table>

**Returns**

Returns a GearsBuilder object with a new template type.

**Example**

```java
GearsBuilder.CreateGearsBuilder(reader).
    localAccumulateBy(r->{
        return r.getStringVal();
    },(k, a, r)->{
        Integer ret = null;
        if(a == null) {
            ret = 0;
        }else {
            ret = (Integer)a;
        }
        return ret + 1;
    });
```