

LengthBased ExampleSelector

This ExampleSelector selects which examples to use based on length. This is useful when you are worried about constructing a prompt that will go over the length of the context window. For longer inputs, it will select fewer examples to include, while for shorter inputs it will select more.

```
from langchain.prompts import PromptTemplate
from langchain.prompts import FewShotPromptTemplate
from langchain.prompts.example_selector import LengthBasedExampleSelector
```

```
# These are a lot of examples of a pretend task of creating antonyms.
examples = [
    {"input": "happy", "output": "sad"},
    {"input": "tall", "output": "short"},
    {"input": "energetic", "output": "lethargic"},
    {"input": "sunny", "output": "gloomy"},
    {"input": "windy", "output": "calm"},
]
```

```
example_prompt = PromptTemplate(
    input_variables=["input", "output"],
    template="Input: {input}\nOutput: {output}",
)
example_selector = LengthBasedExampleSelector(
    # These are the examples it has available to choose from.
    examples=examples,
    # This is the PromptTemplate being used to format the examples.
    example_prompt=example_prompt,
    # This is the maximum length that the formatted examples should be.
    # Length is measured by the get_text_length function below.
    max_length=25,
    # This is the function used to get the length of a string, which is used
    # to determine which examples to include. It is commented out because
    # it is provided as a default value if none is specified.
    # get_text_length: Callable[[str], int] = lambda x: len(re.split("\n| ", x))
)
dynamic_prompt = FewShotPromptTemplate(
    # We provide an ExampleSelector instead of examples.
    example_selector=example_selector,
    example_prompt=example_prompt,
    prefix="Give the antonym of every input",
    suffix="Input: {adjective}\nOutput:",
)
```

[Skip to main content](#)

```
    input_variables=["adjective"],  
)
```

```
# An example with small input, so it selects all examples.  
print(dynamic_prompt.format(adjective="big"))
```

Give the antonym of every input

Input: happy

Output: sad

Input: tall

Output: short

Input: energetic

Output: lethargic

Input: sunny

Output: gloomy

Input: windy

Output: calm

Input: big

Output:

```
# An example with long input, so it selects only one example.  
long_string = "big and huge and massive and large and gigantic and tall and much  
much much much bigger than everything else"  
print(dynamic_prompt.format(adjective=long_string))
```

Give the antonym of every input

Input: happy

Output: sad

Input: big and huge and massive and large and gigantic and tall and much much much
much much bigger than everything else

Output:

```
# You can add an example to an example selector as well.  
new_example = {"input": "big", "output": "small"}  
dynamic_prompt.example_selector.add_example(new_example)  
print(dynamic_prompt.format(adjective="enthusiastic"))
```

[Skip to main content](#)

Give the antonym of every input

Input: happy

Output: sad

Input: tall

Output: short

Input: energetic

Output: lethargic

Input: sunny

Output: gloomy

Input: windy

Output: calm

Input: big

Output: small

Input: enthusiastic

Output: