Data model

Difficulty	
Skills	

Description

You have an existing Pydantic model (`Company`) which reads in a python dict and validates the inputs received.

```
from pydantic import BaseModel, ValidationError
from typing import Optional
from typing_extensions import Literal


#constants
VALUE_1 = "1-9"
VALUE_2 = "10-99"
VALUE_3 = "99+"
VALUE_4 = "unknown"


class Company(BaseModel):
    name: str
    employees: Optional[Literal[VALUE_1, VALUE_2, VALUE_3, VALUE_4]]


if __name__ == '__main__':
    data = {'name': 'Good Company B.V.', 'employees': '1-9'}

    try:
        company = Company(**data)
        print(f"{company.name} has {company.employees} number of employees")
    except ValidationError:
        print(f"Invalid data supplied")
        raise
```

The output of this script is `Good Company B.V. has 1-9 employees`, as expected.

Now, your data provider suddenly changes the data input to this `{'name': 'Amazing Goods B.V.', 'employees': '1'}` 😰 and your code breaks raising a ValidationError.

Being a good engineer that you are, you need to quickly extend the `Company` model to accept below sample inputs to the model without breaking the code.

Sample `data` inputs

You can test your solution with the following inputs:

```
data = {'name': 'Random company A', 'employees': '1'}
data = {'name': 'Random company B', 'employees': '67'}
data = {'name': 'Random company C', 'employees': '101'}
data = {'name': 'Random company D', 'employees': ' 878'} # the whitespace in the stringed number is deliberate !
data = {'name': 'Random company E', 'employees': '0'}
data = {'name': 'Random company F', 'employees': 6}
```

Expected outputs

For each sample input above, the expected output is:

```
>>> "Random company A has 1-9 number of employees"
>>> "Random company B has 10-99 number of employees"
>>> "Random company C has 99+ number of employees"
>>> "Random company D has 99+ number of employees"
>>> "Invalid data supplied"
>>> "Random company F has 1-9 number of employees"
```

Solutions

Provide a possible solution to this problem.

```
from pydantic import BaseModel
from typing import Optional
from typing_extensions import Literal


#constants
VALUE_1 = "1-9"
VALUE_2 = "10-99"
VALUE_3 = "99+"
VALUE_4 = "unknown"


class Company(BaseModel):
    name: str
    employees: Optional[Literal[VALUE_1, VALUE_2, VALUE_3, VALUE_4]]

    # extend here
```

```
if __name__ == '__main__':
    data = {'name': 'Good Company B.V.', 'employees': '1-9'}

    try:
        company = Company(**data)
        print(f"{company.name} has {company.employees} number of employees")
    except ValidationError:
        print(f"Invalid data supplied")
        raise
```

[Before the interview, please send this python script with us via a GitHub repo]