Form filler

* Overall solution (大纲)
* ML Blocking (分块)

1. Preparing training data(测试data)
2. Word tokenizing（把词分开）
3. Word embedding（把每行打成数字）
4. Neural network suggestion（NN形式）

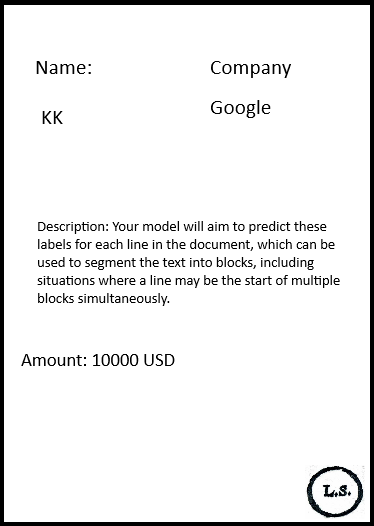
* Word semantic matching (分类)

1. Category/answer split (类别和用户info分开)
2. Fill desired form (填写自己的列表)

**Overall Solution**

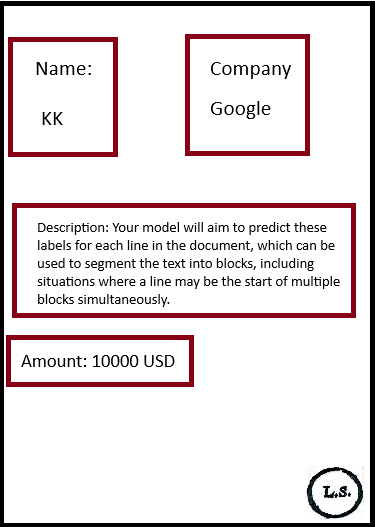
Since the solution requires us to split the given forms into categories, where each block represents a potential category that we want to fill into our own forms, we first need to identify the blocks. Given that the users can provide their own forms in any way they want, we need to train an AI to correctly identify each block. 因为解决方案要求我们将给定的表格分成不同的类别，其中每个块代表我们要填写到我们自己的表格中的潜在类别，我们首先需要识别这些块。鉴于用户可以按照他们想要的方式提供自己的表格，我们需要训练人工智能来正确识别每个块。

Ex:



AI needs to identify each block and correctly associate the name with the name, the company with google, description with description, amount to the right amount. AI需要识别每个区块并正确地将名称与名称、公司与Google、描述与描述、金额与正确的金额相关联。

Result after running the AI:



After extracting each block correctly, information from each block needs to be flattened from top to down. 把block里面的东西打成1D

For example:

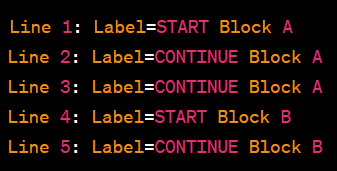


The result will be written into our own file in the correct category

**ML Blocking**

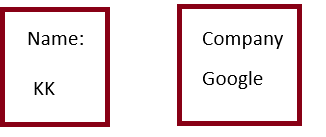
1. **Preparing training data**

Since we are trying to block the forms given, the result needs to be represented in a way that the computer can understand. （output 需要电脑可以理解）



Labels will represent where the block is, the lifetime of a block will be represented by labels start and continue. (label 解释block在哪里，block就是label start 和 continue 来表达)

So if we have the case of



Where the name and company are on the same line,

We would represent it as   
Line 1: Label = Start Block A, Label = Start Block B

Line 2: Label = Continue Block A, Label = Continue Block B

The training data will have to be provided by a human. （这training data 只有人可以提供）

1. **Word Tokenizing**

The input to a machine needs to be numerical, or else it is not going to understand the data. So we first take the form and turn it into lines of strings. Then, the strings will need to be split into tokens, each token representing a word. 把所有的input全部用数字来表达，先得把单词分开

Ex:

Name | Company

KK| Google

1. **Word Embedding**

Name | Company

KK| Google

打成数字用word embedding

Word embedding 会把单词变成 几百度的vector，这个vector就表达这个单词

1. **NN training**

Bi-lstm-crf 模型

Bidirectional lstm conditional random field

Bidirectional：因为这个模型会看以前已经看过的tokens，因它是双向的

Crf 是一个层次，它帮我们看旁边的labels

把模型造好，input是分好的一行一行的 word embedding，output是描述每个block在什么位置

**NN层次详细解释**

1. 输入层（input）输入层接受文本数据的数值表示作为模型的输入。
2. 向量空间（Embedding layer）就是用word embedding 算出来的 2D vector
3. Rnn 层：捕捉依赖关系
4. 双向Rnn 层：考虑看过的和没看过的内容
5. Dense 全连接层：猜测和分类，based on 以前的错和对
6. Crf 层： 序列猜测，sequence prediction
7. Loss：打回神经，告诉神经网络哪里需要修改

**Word Semantics Matching**

1. 类别标签和用户答案分开

* 先打平



* 再tokenize，用nlp

 分成 name 和 kk

* 用word embedding 来研究词性

Name 和 kk 分别打成几百度的数字vector

再查看name， kk 和我们需要match的categories的相似度

比如说：

Name 和 我们需要的“卖家名称” 打成 200度的vector有80%的相似度，那 name 就是 category，后面的就是answer

1. 填我们的表格