
Creative Engine

J. Jason Bell

June 10, 2014

I created a small and simple piece of software which I am calling Creative Engine. Hopefully, Creative Engine can accomplish 3 things:

1. Allow one person to access the knowledge base of hundreds of people for creative purposes
2. Decrease the cost of idea generation
3. Increase the scope of the user's consideration using randomization

The rest of this document is to record progress on the software. I hope to discover drawbacks and then overcome those drawbacks using iteration. The iterations will appear below.

ITERATION 1

The file `blender.R` defines two functions on two lists. The two functions are `blend` and `translate`. The first function should probably be named `apply`, because it just suggests concepts to apply to the target. The `translate` function uses Barry Nalebuff's question "Can you flip it?" with many different operations, in addition to `flip`. I think Nalebuff actually used the word 'flip' to mean more than just a binary switch or a physical rotation of some sort, so `translate` is basically an attempt at a brute-force implementation of that process.

I've developed some intuition about Creative Engine. I believe that an increase in the volume of ideas requires an increase in the sophistication and power of filters. Most ideas are bad and need to be weeded out. That is easier for humans to do when they are dealing with a

few ideas. Creative Engine could produce potentially hundreds of them out in a short time. I believe that the filters people will use are those that come to mind readily while using the software. Things that don't make sense will be tossed away. Perhaps this is a very good way. However, some categorization scheme might help. For example, you might run the code like this:

```
translate("angry birds", category = "apps")
```

That might help people avoid too many translations or concepts that don't apply. On the other hand, just moving on to the next one may be low cost. Filtering rules can be applied *a priori* like this categorization scheme, or *a posteriori*, like the way people throw ideas out that they don't like.

What are some good filtering rules? Survival of the fittest might take too long or cost too much. Humans might make flawed judgements. These are *a posteriori* filtering rules. The categorization scheme might work, though it does slightly compromise purpose #3 of Creative Engine. It seems all *a priori* schemes have this drawback.