Coding Challenge

We've put together this coding challenge to give you a sense of the type of challenges we tackle at Sortable. Impress us! Use any language you'd like as long as we can compile and run it on Linux systems — no Windows stuff, sorry! We're looking for practical solutions with great precision and recall.

Background

At Sortable we do a lot of data integration. A simplified example: product specifications might come from one data source, product pricing from another, and product reviews from a third. A major challenge in working with all this data is determining when two pieces of information from disparate data sources are actually talking about the same product. In academic circles this problem is sometimes called Record Linkage, Entity Resolution, Reference Reconciliation, or a host of other fancy terms. We describe this problem very technically as "matching."

The Challenge

The goal is to use text similarity to match product listings from a 3rd party retailer, e.g. "Nikon D90 12.3MP Digital SLR Camera (Body Only)" against a set of known products, e.g. "Nikon D90".

We'll provide you with a set of products and a set of price listings matching some of those products. The task is to match each listing to the correct product. Precision is critical. We much prefer missed matches (lower recall) over incorrect matches, so try

hard to avoid false positives. A single price listing may match at most one product.

How to enter

- **Complete the challenge in any language you like (that we can build and run on Linux)**
- **Make sure your output is valid JSON lines and conforms to our spec below**
- **Put the source code for your solution up on your GitHub account**
- **Make it easy for us to run your solution (on Linux)**
  - **e.g. "clone my repository and run go.sh"**

- **Email jobs@sortable.com with the location of your GitHub repo and any other relevant info**

What we're looking for

- **Readable, efficient code**
- **Precision – do you make (m)any false matches?**
- **Recall – how many correct matches did you make?**
- **Appropriate data structure and algorithm choices**

The inputs and outputs for this challenge are in the so-called JSON lines format. That is, one valid JSON object per line, with newline separators.

Details

Data Objects

Product

```
{
  "product_name": String   // A unique id for the product
  "manufacturer": String
```

```
    "family": String         // optional grouping of products
    "model": String
    "announced-date": String // ISO-8601 formatted date string, e.g. 2011-04-28T19:00:00.000-
  }
```

Listing

sortable

Ad Optimization **Free Ad Ops Analysis** **Contact Us**

```
  }
```

Result

A file full of Result objects is what your solution will be generating. A
associates a Product with a list of matching Listing objects.

```
  {
    "product_name": String
    "listings": Array[Listing]
  }
```

Challenge Data

Contains two files:

1. **products.txt – Contains around 700 products**
2. **listings.txt – Contains about 20,000 product listings**

Download the data.

Input Files

Products file (products.txt)

> **Text file with one Product object per line.**

Listings file (listings.txt)

> **Text file with one Listing object per line.**

Output File

The output your solution creates should be a text file with one Result object per line.

More about jobs at Sortable.

Sortable is Making Ads Suck Less

Why I bought my company back

Header Bidding Trial by Fire: How 6 Popular Header Bidders Perform

7 Ad Ops Conferences that every Publisher should attend

Ad Mediation: Bridging the Gap Between Publishers and Demand Partners