Building an inductive coding service for employee feedback
Why text?

Text, analysed correctly can identify be a real driver for meaningful change. Our approach means we regularly find themes which are unlikely to have been included in a traditional survey.

“Change the music. It’s so moody and depressing and totally not our customer. We have had several complaints from customers and the team”

in the next survey

“Since the last survey, they have changed the music and it creates a better work place for staff and customers”
<table>
<thead>
<tr>
<th>Question</th>
<th>Question</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>“How can we improve customer experience in our stores?”</td>
<td>“What could we do to improve the performance review process?”</td>
<td>“What could we do to improve your working life here?”</td>
</tr>
<tr>
<td>“What ideas do you have to improve the Christmas sales period?”</td>
<td>“How could HR improve the services they provide you as a manager?”</td>
<td>“If you were our Brand Director, what would you do first?”</td>
</tr>
</tbody>
</table>
Finding meaningful themes

Ideally the theme labels should be themselves reasonable answers to the question.

More clarity on the future direction
Improve performance review process
Simplify sales plans
Solve issues related to stock
More Development & Training Required
Proud of our Heritage
Improved Discussion of Results Required
Part-Time vs. Full-Time
Better Planning Required
More Focus On Performance Needed
Better Home Office Options Required
Lack of Opportunities/Chances
Better Customer Service
Mindset/Mentality Change Needed
Key Challenges

In most instances our real competitor is “doing nothing.”

- Sentence fragments & spelling
- Multiple languages
- Medium-sized datasets
- Turnaround needs to be quick
- Can’t cost a fortune
Specific Models

> 

Generic Models
( Humans + Algorithms )

> 

( Humans OR Algorithms )
People in organizations develop their own language

{eg. Project Names, Acronyms, just plain misuse...}
Given our constraints, there are two things that define our success:

- Optimising on two variables
- Quality
- Human time
The expense of creating labelled data defines how we tackle the problem.

<table>
<thead>
<tr>
<th>Mechanical Turk</th>
<th>Our Coders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assume that coding is cheap</td>
<td>Assume that coding is expensive</td>
</tr>
<tr>
<td>Non-specialised coding</td>
<td>Domain Experts needed</td>
</tr>
<tr>
<td>Data is ‘open’</td>
<td>Confidential Data</td>
</tr>
<tr>
<td>Tasks are provided uniformly</td>
<td>Provide tasks based on expected marginal value</td>
</tr>
</tbody>
</table>
Our Process

A highly-simplified representation of our overall text coding process.

Pre-processing → Unsupervised Clustering → Expanding rule set → Automated Semi-Supervised learning → Supervised Learning

Active Learning

Pattern spotting with themes

Analysis of ‘Not Coded’

What themes are present?

E.g. using translation-loops to increase volume of training data
Maximising human input

A few examples

Healthy

Where to focus

More work needed

Overfit

Possibly linked to another concept
Coding tools - Buy vs Build

We decided to build our own coding platform

- Provision domain experts to the right projects
- Data Scientist out of the loop
- Additional flexibility - eg visualisation
- Measure productivity
- Cost
Most clients ask us to identify additional insight from their responses, either looking just at the text data or combining it with other employee information.
Summarising underlying text

We use proprietary document summarisation algorithms to identify example sentences
Some things have worked better than others

- Domain-specific embeddings
- Refine before automating
- Experiment, experiment, experiment
What next?

There’s always too many ideas to try!

- Improve automatic labelling
- Better models for uncertainty (which to pass to human)
- Improve tools for coders (work on UI / UX)
- Building generic models for rare but important themes