In [17]:
```
import scattertext as st
import re, io
from pprint import pprint
import pandas as pd
import numpy as np
from scipy.stats import rankdata, hmean, norm
import spacy
import os, pkgutil, json, urllib
from urllib.request import urlopen
from IPython.display import IFrame
from IPython.core.display import display, HTML
from scattertext import CorpusFromPandas, produce_scattertext_explorer
display(HTML("<style>.container { width:98% !important; }</style>"))
```

In [18]:
```
data = pd.read_csv('reviews.csv')
data = data[['reviews','stars']]
```

In [32]:
```
data['sentiment'] = ['positive' if x in [4,5] else 'negative' for x in dat
```

In [25]:
```
nlp = spacy.load('en')
```

In [54]:
```
corpus = st.CorpusFromPandas(data, category_col='sentiment', text_col='rev
```

In [60]:
```
# Terms that appear more frequently in the reviews corpus than any other F

list(corpus.get_scaled_f_scores_vs_background().index[:10])
```

Out[60]:
```
['wex',
 'parcelforce',
 'phoned',
 'knowledgable',
 'hesitation',
 'knowledgeable',
 'dslr',
 'packaged',
 'faultless',
 'arrived']
```

In [68]:
```
# Terms most associated with positive reviews

term_freq_df = corpus.get_term_freq_df()
term_freq_df['positive_sentiment'] = corpus.get_scaled_f_scores('positive'
list(term_freq_df.sort_values(by='positive_sentiment', ascending=False).in
```

Out[68]:
```
['highly',
 'as always',
 'excellent service',
 'quick delivery',
 'recommend',
 'fast delivery',
 'excellent',
 'would recommend',
 'fast',
 'easy to',
 'well packaged',
 'recommended',
 'quick',
 'service as',
 'recommend wex',
 'competitive',
 'knowledgeable',
 'efficient',
 'well packed',
 'brilliant',
 'first class',
 'great service',
 'reliable',
 'prompt',
 'easy',
 'as described',
 'usual',
 'as usual',
 'class',
 'always']
```

In [70]:
```
# Terms most associated with negative reviews

term_freq_df['negative_sentiment'] = corpus.get_scaled_f_scores('negative'
list(term_freq_df.sort_values(by='negative_sentiment', ascending=False).in
```

Out[70]:
```
['was told',
 'told',
 'poor',
 'manager',
 'until',
 'refund',
 'reply',
 'send',
 'an email',
 'weeks',
 'still',
 'waiting',
 'left',
 'barclays',
 'they would',
 'saying',
 'email',
 'paid for',
 'pounds',
 'had been',
 'told that',
 'said',
 'paid']
```

In [90]:
```
data.head()
```

Out[90]:

	reviews	stars	parsed	sentiment
0	Tel order to Wex for Atomos Shinobi. Little p...	1	(Tel, order, to, Wex, for, Atomos, Shinobi, .,...	negative
1	Service was good fast delivery, the rwasi. For...	1	(Service, was, good, fast, delivery, ,, the, r...	negative
2	Since the take over of Calumet, Wex has failed...	1	(Since, the, take, over, of, Calumet, ,, Wex, ...	negative
3	I returned a camera to them unused and they wo...	1	(I, returned, a, camera, to, them, unused, and...	negative
4	The worst customer service I have ever experie...	1	(The, worst, customer, service, I, have, ever,...	negative

In [97]:
```
html = produce_scattertext_explorer(corpus,
                                    category='sentiment',
                                    category_name='positive',
                                    not_category_name='negative',
                                    width_in_pixels=1000,
                                    minimum_term_frequency=5,
                                    term_significance = st.LogOddsRatioUni
                                    include_term_category_counts=False)
file_name = 'test.html'
open(file_name, 'wb').write(html.encode('utf-8'))
IFrame(src=file_name, width = 1200, height=700)
```

```
---------------------------------------------------------------------
----
AssertionError                            Traceback (most recent call l
ast)
<ipython-input-97-4548104c9897> in <module>()
      7                                         term_significance = st.LogO
ddsRatioUninformativeDirichletPrior(),
      8                                         metadata = np.array([x for
 x in data['stars']]),
----> 9                                         include_term_category_count
s=False)
     10 file_name = 'test.html'
     11 open(file_name, 'wb').write(html.encode('utf-8'))

~/anaconda3/lib/python3.6/site-packages/scattertext/__init__.py in prod
uce_scattertext_explorer(corpus, category, category_name, not_category_
name, protocol, pmi_threshold_coefficient, minimum_term_frequency, mini
mum_not_category_term_frequency, max_terms, filter_unigrams, height_in_
pixels, width_in_pixels, max_snippets, max_docs_per_category, metadata,
scores, x_coords, y_coords, original_x, original_y, rescale_x, rescale_
y, singleScoreMode, sort_by_dist, reverse_sort_scores_for_not_category,
use_full_doc, transform, jitter, gray_zero_scores, term_ranker, asian_m
ode, use_non_text_features, show_top_terms, show_characteristic, word_v
ec_use_p_vals, max_p_val, p_value_colors, term_significance, save_svg_b
utton, x_label, y_label, d3_url, d3_scale_chromatic_url, pmi_filter_thr
esold, alternative_text_field, terms_to_include, semiotic_square, num_t
erms_semiotic_square, not_categories, neutral_categories, extra_categor
ies, show_neutral, neutral_category_name, get_tooltip_content, x_axis_v
alues, y_axis_values, color_func, term_scorer, show_axes, horizontal_li
ne_y_position, vertical_line_x_position, show_cross_axes, show_extra, e
xtra_category_name, censor_points, center_label_over_points, x_axis_lab
els, y_axis_labels, topic_model_term_lists, topic_model_preview_size, m
etadata_descriptions, vertical_lines, characteristic_scorer, term_color
s, unified_context, show_category_headings, include_term_category_count
s, div_name, alternative_term_func, return_data)
    446                                                     extra_c
ategories=extra_categories,
    447                                                     backgro
und_scorer=characteristic_scorer,
--> 448                                                     include
_term_category_counts=include_term_category_counts)
    449     if return_data:
    450         return scatter_chart_data

~/anaconda3/lib/python3.6/site-packages/scattertext/ScatterChartExplore
```

```
r.py in to_dict(self, category, category_name, not_category_name, score
s, metadata, max_docs_per_category, transform, alternative_text_field,
 title_case_names, not_categories, neutral_categories, extra_categorie
s, neutral_category_name, extra_category_name, background_scorer, inclu
de_term_category_counts)
    108                                             neutral_categories=neu
tral_categories,
    109                                             extra_categories=extra
_categories,
--> 110                                             background_scorer=back
ground_scorer)
    111         docs_getter = self._make_docs_getter(max_docs_per_categ
ory, alternative_text_field)
    112         if neutral_category_name is None:

~/anaconda3/lib/python3.6/site-packages/scattertext/ScatterChart.py in
to_dict(self, category, category_name, not_category_name, scores, trans
form, title_case_names, not_categories, neutral_categories, extra_categ
ories, background_scorer)
    266
    267         all_categories = self.term_doc_matrix.get_categories()
--> 268         assert category in all_categories
    269
    270         if not_categories is None:

AssertionError:
```

In [30]:
```
def normcdf(x):
    return norm.cdf(x, x.mean(), x.std())
```

In [ ]:
```

```