Discovering domain-sensitive topics in user-reviews

Sarthak Kamat

Abstract
Customer reviews are integral to retail businesses. This paper demonstrates new methods for ranking the most representative and interesting snippets within reviews posted by their customers.

Definition
Review: a single user-posted review usually about an interaction with a (small) business.
Domain: the set of all reviews for a business. The domain is sometimes called a "review-set".
Phrase: a small section of a review.
Snippet: a tuple containing: one topical keyword, descriptors used for that keyword, sentiments of the descriptor. The snippet is the canonical form of a phrase.

Overview
Step 1. Break down a review phrase into its canonical form snippet := (keyword, descriptor, sentiment).
Step 2. Define a score function for a keyword within its domain (which maps 1:1 to a business).
Step 3. Aggregate the score function over all the keywords.
Step 4. Find the most talked-about keywords within a business for a particular sentiment.
Step 5. Score and find out the best descriptors for the set of keywords.

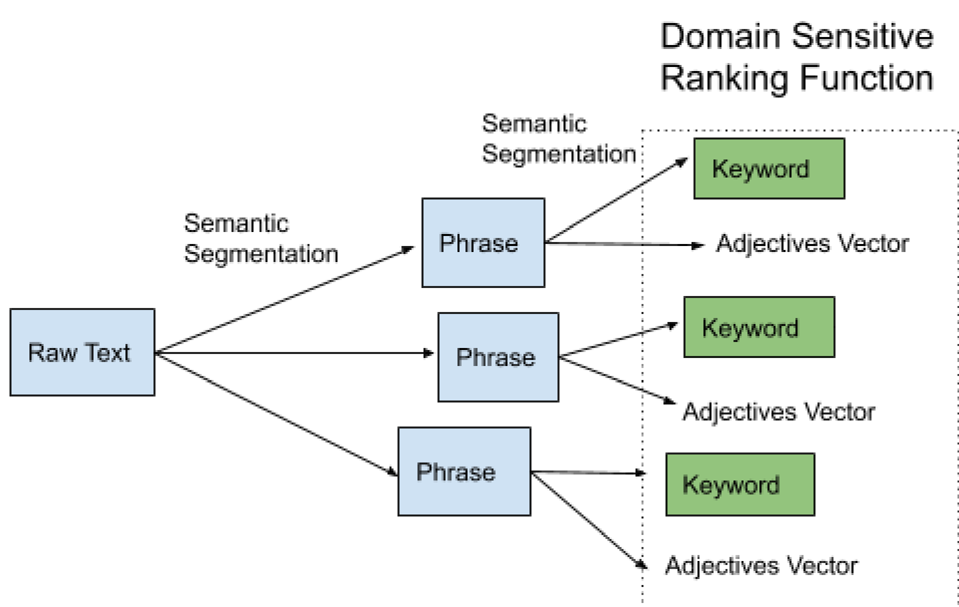

Figure 1: Architecture Overview. The text is broken into phrases, then keywords and adjectives, then ranked with a domain sensitive function.

Method

The first thing to consider was a formal definition of two snippets conveying the same information. We are focusing on Steps 2-5 in this paper. Incidentally, Step 1 is not the focus of the method in this paper, for which there's a proprietary algorithm to identify keywords, their corresponding descriptors, and their sentiments. We are using A_Bridal_Business as the example business for the rest of the paper.

So, a review that says:

> "Staff was kind of rude.. I minused a star because the staff was just that annoying. Regardless, I didn't care about his opinion. But he did give us a great discount on my engagement ring. I ended up getting a 14 karat rose gold engagement ring and I'm soooo in love with how beautiful it looks."

Is restructured as:

```
["staff", ("rude":-1)]
["staff", ("annoying":-1)]
["discount",("great":1)]
["engagement ring",("14 carat:1, "rose gold":1)]
["looks",("love":1,"beautiful":1))]
```

(where 1 implies a positive snippet and -1 implies a negative snippet)

The main strategy is to rank all the topical keywords over a domain of reviews, which in this case is an entire business, and then pick the highest ranking descriptor used with that keyword using the same scoring technique.

The simplest approach would be to define a scoring function as:

$t_i \in T$, the set of all topical keywords

$t_k \in D_j$, the set of keywords within a domain D_j

$t_m \in R_n$, the set of keywords within a review R_n

$ScoreTF_{ij}$: (T X Domain D) -> #(T U D)

Where #A denotes the cardinality of any set A. This function is commonly known as the term-frequency score or TF score.

Then we use the method ScoreTF for all descriptors with each of the top keywords and find the best descriptor for that keyword.

Here are the results for applying top10(ScoreTF) over A_Bridal_Business

1. **“great experience”** in 8340 reviews
2. **“wonderful experience”** in 3033
3. **“amazing experience”** in 2051
4. **“good experience”** in 1524
5. **“great service**” in 1611
6. **“great job”** in 1593
7. **“best experience”**: 1460
8. “**perfect dress**” in 1410
9. **“friendly staff**” in 1374
10. “**helpful staff**” in 1358

Figure 2: Standard rankings by TF score above. Red indicates an uninteresting snippet, green indicates an interesting snippet.

Although this method did work, there are certain problems it does not address.

> 1. The primary one being that most of the keywords are not very descriptive or helpful. The reason is that the keywords are not domain specific -- words like experience, staff, and job do not necessarily relate to the product line of A_Bridal_Business
>
> 2. Same goes for descriptors. Most of the descriptors like great, good, amazing, wonderful do convey emotion, but not specific or interesting emotion.

To resolve this, we must recognize that not all keywords must be scored the same way simply because they occur most often (as $\mathrm{ScoreTF}$ would do).

There must be weightage for the relevance of a particular keyword to that domain, and the way to that is to use a concept similar to inverse-document frequency (IDF) of that keyword.

The Domain Sensitivity of a topical keyword is defined as follows:

$$\mathrm{DS}(\mathrm{TopicalKeyword}\ t_i) := \log\left(\frac{\#R}{\#(R \cap t_i)}\right)$$

Where the numerator is the total number of reviews across all domains, and the denominator is the total number of reviews containing the topical keyword t_i

DS measures the ratio of the number of reviews to the reviews containing k, thus demoting keywords that are highly common across all domains (such as *experience*, that would arise in all businesses) and promoting keywords that are more common in this domain (such as *dress* or *ring* that would have arise less in other businesses and more within A_Bridal_Business) .

So we can redefine score as the scalar product of the TF score and DS score, weighed further by (α for TF, and β for DS). The domain-sensitive score can be defined as:

$$DSS_{ij} : (\mathrm{T\ X\ D}) \rightarrow ScoreTF^{\alpha}_{ij} \cdot DS_i^{\beta}$$

Once DSS is used to determine the top keywords, we can similarly use DS and DSS() on the set of descriptors that use that keyword. So the descriptors that are used a lot for all keywords will be demoted (such as the descriptor *great* for the world *label*), whereas the descriptors that are used more for this keyword will be promoted (such as the descriptor *tasteful* for the keyword *label*).

Results

Presetting $\alpha = 0.5$ and $\beta = 2$ on top10(DSS) over A_Bridal_Business, these were the improved results:

1. "**perfect dress**" in 1410 reviews
2. "**amazing alteration**" in 167
3. "**helpful stylist**" in 234
4. "**easy process**" in 76
5. "**correct size**" in 128
6. "**so much fun**" in 945
7. **"tasteful label**" in 186
8. "**nice person**" in 353
9. **"perfect bridesmaid dress**" in 99
10. **"great value**" in 178

Figure 3: Rankings by domain sensitive model gives much more detailed and interesting results. Red indicates an uninteresting snippet, green indicates an interesting snippet.

We can see the general improvement in quality of results through the creation of a separation between keywords and adjectives and assessment of domain sensitivity.