Informationsextraherare – Ölrecensioner

Hugo Forss och Henning Norén

1 Introduktion

1.1 Prolog

När det här projektet påbörjades så var vårt syftet att skriva en informationsextraherare. Att det blev för just ölrecensioner beror främst på två aspekter. Dels var det ett personligt intresse - det var en chans att lära sig mer om ölkultur och att få utveckla ett verktyg som faktiskt kunde vara användbart. Dels fanns det en begränsat mängd information att hämta och en begränsad vokabulär (en sanning med viss modifikation).

Det första problem vi ställdes inför var att finna en korpus som var stor nog. Vad vi fann var en sida vid namn Sidan RateBeer.com. Den kan i princip ses som en gigantisk databas med ölrecensioner, skrivna av sidans medlemmar, sorterade under respektive öl. Att det därmed redan framgår vilket öl det är frågan om gjorde att vi fick mindre information att försöka plocka fram. Å andra sidan gav det oss material nog att prova en annan idé - att slå samman informationen från de olika recensionerna och få fram den allmänna uppfattningen om varje öl (och i slutänden få fram något liknande posterna i Systembolagets kataloger).

Vidare gav sidan mycket nyttig information i form av artiklar om ölprovning för nybörjare samt en nybörjarvokabulär som för oss utgjorde en ypperlig startpunkt. Vad vi genast noterade var att vokabulären var uppdelad enligt 4 olika kategorier. Då många av recensionerna snyggt och prydligt tog upp kategorierna en efter en föll det sig naturligt att försöka fånga in hela längre fragment innehållande ord tillhörande samma kategori.

1.2 Öltermer

Ett öl bedöms efter fyra egenskaper: utseende, arom, smak och gomkänsla. Utseende handlar om färgen på vätska och skum, huruvida vätskan är klar eller grumlig och om skummet är tjockt och långlivat eller inte. De tre smakerna är sött, surt eller beskt (öl brukar i regel inte vara salt). Arom är det vi tar upp med näsan och brukar allt som oftast kallas för smak det också. Här handlar det i första hand om att fri association och därför finns det också en närmast obegränsad mängd ord inom denna kategori. Gomkänsla är slutligen huruvida ölet upplevs som torrt, oljigt, stickigt med mera.

2 Informationsextrahering

2.1 Förberedelser

Ett Perl-script loggar in på sidan RateBeer.com och hämtar samtliga recensioner för ett givet öl baserat på dess id-nummer. Därefter snyggar ett annat script recensionerna och utför inledande taggning.

2.2 Regler

Parsning och taggning sköts av SloppyTagger. Redan i ett tidigt stadium bestämde vi oss för att inte använda ordklasstaggning. Istället arbetar taggern i flera pass där den matchar och taggar allt mer komplexare strukturer. Reglerna är skrivna med en regex-inspirerad syntax som låter oss matcha både klartext och xml-taggar samt att infoga nya taggar på valfri plats i de matchade sekvenserna.

2.3 Ord

I det första passet taggas de ord som vi har definierat i vår ordlista. Dessa kan delas in i fyra kategorier:

- Kategoriord är ett ord eller mönster som anger kontext.
- Nyckelord är ord som bär på information. De flesta nyckelord kan klassificeras direkt i ordlistan, men några - speciellt färg - blir helt beroende av kontext.
- Modifierare är ord som (förstärker/försvagar/inverterar) ett nyckelords betydelse. De kan också användas för att hitta okända nyckelord. Modifierare är egentligen väldigt svåra då det för det mesta inte går att gissa recensentens intention. Eftersom vi inte vet om en modifierare syftar på ett eller flera ord så delar vi helt enkelt in dem i två kategorier, de som står före och de som står efter sitt nyckelord.
- Länkord är alla typer av bindeord. De hjälper ytterligare vid kontextbestämning.

En lockande tanke vore att helt slopa ordlistan så när som på kategoriorden och därmed låta programmet själv identifiera nyckelord. Men så som informationsextraheraren ursprungligen var tänkt att fungera behövde vi kunna ange betydelser för flertalet ord (som modifierarnas vikter).

I vissa fall finns en så pass liten och väl vedertagen vokabulär att vi bedömt det onödigt att leta efter ytterligare uttryck. Ett annat skäl kan också vara att det råder en tydlig relation mellan de olika orden. Färger är ett tydligt exempel och modifierare än mer så. I vår ursprungliga design placeras dessa på en slags skala och i slutänden är det tänkt att informationsextraheraren ska presentera ett slags genomsnitt av dessa värden.

2.4 Enheter

Nyckelord och modifierare bildar tillsammans enheter där nyckelordet anger grundbetydelse och modifieraren ordets vikt. Enheterna förenklar på många sätt kommande steg. Dels ger det oss möjligheten att placera nyckelord tillhörande mer specifika kategorier i enheter med mer generella namn. Med detta uppnår vi att vi på ett enkelt sätt kan matcha ord från olika kategorier med en gemensam regel.

Med enheternas hjälp kan vi också tagga kringliggande ord. Många gånger är recensionerna byggda som uppräkningar av olika egenskaper och därför är det naturligt att misstänka att intilliggande ord också kan vara nyckelord.

2.5 Fragment

Fragment är ytterligare ett sätt att förstärka ordens kontext. Här taggas sammanhängande sjok av redan taggade ord som ofta inleds eller avslutas med ett kategoriord. Dessutom hänger vissa kategorier naturligt samman och återfinns ofta tillsammans åtskilda av ett länkord. Andra separeras mer naturligt av meningsgränserna.

Dessa regler kommer inte att finna några nya nyckelord, tvärtom används de för att plocka bort enheter som inte ingår i något fragment och som därför kan ha taggats felaktigt. Fragmenten används också för att lösa vissa oklarheter. Exempelvis genom att skilja de färger som beskriver ölets kropp (själva vätskan) från de som beskriver dess skum.

2.6 Sammanslagning

Ett problem med språk som engelska är dess många böjnings- och avledningsformer. På något vis ska de olika formerna räknas samman. Enklast är att utnyttja storleken hos vår textmassa. I en speciell ordlista finns ett antal kända ändelser angivna. Orden matchas mot de olika ändelserna som isåfall plockas bort/ersätts med sin grundform. Det nya ’hypotetiska’ ordet testas därefter mot textmassan för att se om det förekommer någonstans.

Av de ord som taggats och klassificerats väljs de mest frekventa ut. På så sätt kommer en stor del av alla felklassificeringar att sorteras bort.

3 Utvärdering

För att kunna bedöma hur pass väl informationsextraheraren fungerar så har vi för hand plockat ut information ur ett flertal öl. Två jämförelser görs: dels övergripande proportionerna falska positiver och negativer, dels hur många av de topprankade nyckelorden den lyckas pricka rätt.

Ovanstående tabell visar ration för antalet falska negativer (sådana vi inte taggat men som vi borde) samt falska positiver (sådana vi felaktigt taggat) för testet mot våran testmängd. Fler falska negativer än positiver tyder på att våra regler är alltför strikta och borde skrivas mer generella. Kategorierna palate, head och body lider av att det är rätt få recensenter som tar upp dem i sina recensioner. De kan också ses som mer abstrakta än de andra kategorierna. Vad färg, smak och arom beträffar så brukar recensenterna vara betydligt mer ense.

Kategori	Falska Negativa	Falska Positiva
AROMA	0.602	0.061
FLAVOUR	0.767	0.029
PALATE	0.388	0.511
HEAD	1.289	0.130
BODY	1.308	0.111
HEADCOLOUR	0.256	0.400
BODYCOLOUR	0.366	0.516
TOTAL	0.611	0.157

Som man ser så är precisionen inte särskilt upplyftande om man inte plockar fram majoritetsvärdena.

Ovanstående tabell visar på precisionen när vi enbart tar med majoritetsvärdena för kategorierna. Varje kategori har med sitt majoritetselement utom aroma som har med sina 3 största majoritetselement. Ökar man antalet majoritetselement till fyra får vi 92% precision samt 87% för 5. Samma mönster som syntes ovan kommer tydligare fram nu när vi bara har majoritetselementen. Head samt Palate har vi vissa problem med och Body som både är ganska ovanligt i beskrivningarna samt saknar tydliga mönster för beskrivningen har vi väldigt dåligt resultat på.

4 Slutsats

Vi har inte nått fram till vårat slutmål, att producera kort läsbar sammanfattning av ölet men från våra resultat så bör det inte vara några större problem. Det som t ex systembolaget brukar ha med är tre-fyra aromer, färg samt flavour. Dessa visar ovanstående att vi med god sannolikhet kan plocka ut ur en tillräckligt stor mängd recensioner. Det finns dock mer att göra. Vikter för modifierare och färg bör kunna tas in för att ge mer nyans och precision i beskrivningen. Förbättringar av matchning av Body, Head samt Palate bör kunna göras. Vår bristfälliga suffix-hantering borde bytas ut mot en riktigt transducer, anpassad för den typ av något kaotiska data som vi jobbar med, vilket antagligen ökat precisionen ännu mer. Vi anser att med viss anpassning så skulle vårat system kunna köras mot en levande recensionsdatabas för öl och automatiskt plocka fram majoritetsuppfattningen av de mer populära kategorierna med tillräckligt stor precision för att vara intressant.

A Användarhandledning

För att systemet skall fungera måste man skapa två underbibliotek från där man har programmen;
`reviews` - här hamnar alla recensioner man tankar hem, döpta efter ölet. Töm detta bibliotek mellan körningarna eller tag och använd en backup mellan momenten om du vill köra flera gånger då alla programmen utom `comparer.pl` och `stats.pl` är destruktiva och förändrar innehållet i detta bibliotek
`creviews` - här skall alla handtaggade data ligga, döpta efter ölet med `WORKED` i slutet av namnet.

A.1 autologin

`autologin.pl` - autologin är en automatiserad recensionsinsammlare för vår databas (ratebeer.com). Programmet tar ett eller flera argument i form av idn för ölet man vill hämta ner. Man kan även begränsa så att programmet bara hämtar ner den första sidan genom att som första argument använda `-f`. Autologin använder ett hårdkodat användarnamn och lösenord till ratebeer.com men man kan enkelt ändra det genom att ändra variablerna `$username` samt `$password` på rad 12 resp. 13.

A.2 formatreviews

`formatreviews.pl` - formatreviews formaterar, rensar och bygger xml-träd av den råa datan som autologin har tankat hem. Programmet tar inga argument.

A.3 sloppytagger

`sloppytagger.pl` - sloppytagger är själva taggaren som tar som argument filnamnet för en regelsamling och tillämpar den på recensionerna. Generellt kör man detta program i flera pass med

Kategori	Top X element	Andel rätt
AROMA	3	100%
FLAVOUR	1	100%
PALATE	1	50%
HEAD	1	50%
BODY	1	33%
HEADCOLOUR	1	100%
BODYCOLOUR	1	100%

olika regelsamlingar för att på så sätt fånga mer och mer komplexa mönster.

A.4 collector

`collector.pl` - collector samlar ihop data från de taggade recensionerna och stoppar in dem i de olika kategorierna. Formatet är detsamma som de handtaggade recensionerna är i så detta är slutprodukten av själva insamlingen av information. Collector tar inga argument.

A.5 comparer

`comparer.pl` - comparer jämför slutprodukterna från reviews med de handtaggade filerna. Programmet kräver att underbiblioteket `reviews` innehåller alla de recensioner som finns i `creviews`. Programmet tar inga argument och skriver ut resultatet på skärmen, ölvis och kategorivis och sedan ett slutresultat som avslutas med det kryptiska TOP TOTAL. Nästa rad skriver ut hur många ölsorter/filer som behandlats och sedan 7 heltal som representerar de olika kategorierna. Värdena för hur många majoritetselement som skall vara med är hårdkodat till 3 för aroma och 1 för resten och ordningen på dem är: aroma, flavour, palate, head, body, headcolour, bodycolour. Varje heltal representerar hur många majoritetselement som stämmer med de handtaggade och för 100% så skall alltså `antal filer` $\times$ `antal majoritetselement` vara lika med värdet som skrivs ut. En nolla innebär att inga matchade. Progrmmet är inte destruktivt så man kan köra om utan att skydda `reviews` eller `creviews`.

A.6 stats

`stats.pl` - stats skriver ut en sammanfattning för vilka ord som klassats i vilken kategori för varje öl som finns taggat. Programmet tar inga argument och och är inte destruktivt så det kan upprepas utan att man behöver skydda `reviews`

A.7 Exempelkörning 1

Vi plockar hem ett sex stycken ölsorter och kör dem genom hela systemet för att avsluta med att skriva ut en sammanfattning om dem.;

```
  $>./autologin.pl 12492 32111 36624 51539 6115 8484 &&
./formatreviews.pl && ./sloppytagger.pl ordlista  &&
./sloppytagger.pl enheter && ./sloppytagger.pl kandidater &&
./sloppytagger.pl fragment && ./collector.pl && ./stats.pl

<---------- Valley_Brew_Uberhoppy_IPAWORKED ---------->

AROMA: hops(10) malty(7) alcohol(5) caramel(4) toast(3) hoppy(2) resin(2)
woody(1) ginger(1) yeasty(1) lemon(1) grass(1) chocolate(1) floral(1)
citrus(1) fruity(1)
FLAVOUR: sweet(5) bitterness(2) bitter(2) sweetness(1)
PALATE: bodied(3) carbonation(3) thin(1)
HEADCOLOUR: tan(2) white(1) yellow(1)
...
$>_
```

A.8 Exempelkörning 2

Efter att vi kört exempel 1 så vill vi jämföra resultatet mot de handtaggade filer vi har, som råkar stämma precis med de ölsorter vi har tankat hem och jobbat med;

```
$>./compare.pl

<---------- Valley_Brew_Uberhoppy_IPAWORKED ---------->

<--- AROMA --->
MATCH: 30       malt(6) alcohol(4) hops(4) toasty(3) hoppy(2) resin(2)
woody(1) yeasty(1) ginger(1) caramelly(1) grass(1) chocolate(1) floral(1)
citrus(1) fruity(1)
F_NEG: 55       hops(6) hop(4) caramel(4) grapefruit(3) caramelly(3) pine(3)
citrusy(3) malts(2) bready(1) piney(1) milk(1) spice(1) mango(1) tropical(1)
resiniousness(1) amarillo(1) orange(1) cookie(1) gingersnap(1) malt(1)
alcohol(1) peppery(1) grassy(1) flesh(1) blackberryish(1) grassiness(1)
maltiness(1) grapefruity(1) spiciness(1) juiciness(1) berryish(1) cascade(1)
roastiness(1) papaya(1) rye(1)
F_POS: 4        resin(2) lemon(1) citrus(1)
false negative rate: 1.618, false positive rate: 0.047
<--- FLAVOUR --->
MATCH: 7        sweet(3) bitter(2) sweetness(1) bitterness(1)
F_NEG: 3        sweet(2) bitterness(1)
F_POS: 1        bitter(1)
...
<--- BODY --->
false negative rate: 1.308, false positive rate: 0.111
<--- BODYCOLOUR --->
false negative rate: 0.366, false positive rate: 0.516
<--- TOTAL --->
false negative rate: 0.611, false positive rate: 0.157
<--- TOP TOTAL --->
of 6 files:  18 6 3 3 2 6 6
$>_
```