General Architecture for Text Engineering

Module 7:Opinion Mining

Diana Maynard (d.maynard@sheffield.ac.uk)

GATE General Architecture for Text Engineering

This session will be recorded

Recorded video will be available after this session

GATE General Architecture for Text Engineering

Warning:

these slides and hands-on material contain swear words and abusive terms

Aims of this session

- Introduced the concept of Opinion Mining and look at some issues
- Demonstrate simple examples of rule-based and ML methods for creating Opinion Mining applications
- Consider how these can be extended / adapted
  - Examples of how deeper linguistic information can be useful
  - Practice with complex applications
  - Practice with ML

What is Opinion Mining?

- A relatively recent discipline that studies the automatic extraction of opinions from text
- More informally, it's about extracting the opinions or sentiments given in a piece of text
- Also referred to as Sentiment Analysis (these terms are roughly interchangeable)
- Web 2.0 nowadays provides a great medium for people to share things.
- This provides a great source of unstructured information (especially opinions) that may be useful to others (e.g. companies and their rivals, other consumers...)

It's about finding out what people think...

Opinion Mining is Big Business

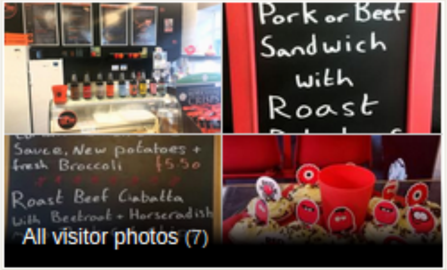
- Someone who wants to buy a camera
  - Looks for comments and reviews
- Someone who just bought a camera
  - Comments on it
  - Writes about their experience
- Camera Manufacturer
  - Gets feedback from customer
  - Improve their products
  - Adjust Marketing Strategies

Cafe Pie

15 Reviews | #24 of 1,084 Restaurants in Sheffield | #26 of 1,169 Places to Eat in Sheffield

£ | Café, British, English

Overview | Reviews (15) | Q&A | Location



TripAdvisor Reviewer Highlights

Read all 15 reviews

Visitor rating

Excellent	14
Very good	1
Average	0
Poor	0
Terrible	0

“Mrs ”

Amazing pies and great friendly staff my kids loved the kids snack packs too excellent idea! I'll definitely be back soon to try a full English breaky!!

Reviewed 6 May 2015

Spring1986

GATE General Architecture for Text Engineering

Café Pie Reviews

Alex P

1 review

"Proper homemade tastey English grub"

Reviewed 3 May 2015 via mobile

Finally went to the pie cafe for lunch yesterday. Had pie mash and red cabbabe, really tasty, pastry was perfect, plenty of filling and red cabbage had loads of flavour. Loved the diner style decor that still stuck to its Yorkshire routes. I'll definately be back for more. Yum yum!

Was this review helpful? Yes

RyuOnline

Reviewer

4 reviews

4 restaurant reviews

"The Famous Cafe Pie in Walkley "

Reviewed 30 April 2015

Superb cafe with great selection of fresh sandwiches & meals. Sunday roasts with pie or meat are to die for. This is not your average greasy vera sandwich shop. They take great pride in all the food they produce from bacon bap to full themed dishes. & with the name of the shop called "Cafe Pie" it's a must to...

More

It's not just about product reviews

- Much opinion mining research has been focused around reviews of films, books, electronics etc.
- But there are many other uses
  - companies want to know what people think
  - finding out political and social opinions and moods
  - investigating how public mood influences the stock market
  - investigating and preserving community memories
  - drawing inferences from social analytics

Some online sentiment analysis tools

- Lexalytics (was Semantria) https://www.lexalytics.com/demo (general)
- TipTop: http://feeltiptop.com/ (tweets)
- Parallel Dots https://www.paralleldots.com/sentiment-analysis (general)
- QuickSearch https://www.talkwalker.com/quick-search-form (brand comparison)
- NCSU Sentiment Viz (general) https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/

GATE General Architecture for Text Engineering

Lexalytics

Let's start by analyzing a single document:

English

No Industry Pack

Enter a URL... | Go

Highlight: ◉ Phrases ○ Themes ○ Entities

Joseph Levy was preparing for a season of scientific research in Antarctica last week when he got the call: Stand down.

Dr. Levy, a research associate at the University of Texas at Austin's Institute for Geophysics, is studying the climate history of the dry valleys of Antarctica by analyzing buried ice sheets that have been frozen since the last ice age and are beginning to thaw.
The research season in Antarctica typically starts around now, when things warm up enough to be merely frigid and scientists from around the world flock far south to conduct studies that affect our understanding of climate change, volcanoes, the family life of Weddell seals and much more. But with the United States government partly shut down, federally financed research has come to a halt for Dr. Levy and hundreds of other Americans. Even if a budget deal is struck, these scientists will have less time on the ice, and some will lose a full year's worth of work as the narrow window of productive time closes.
"It's like a biography of the earth with a couple of pages in the middle torn out," Dr.

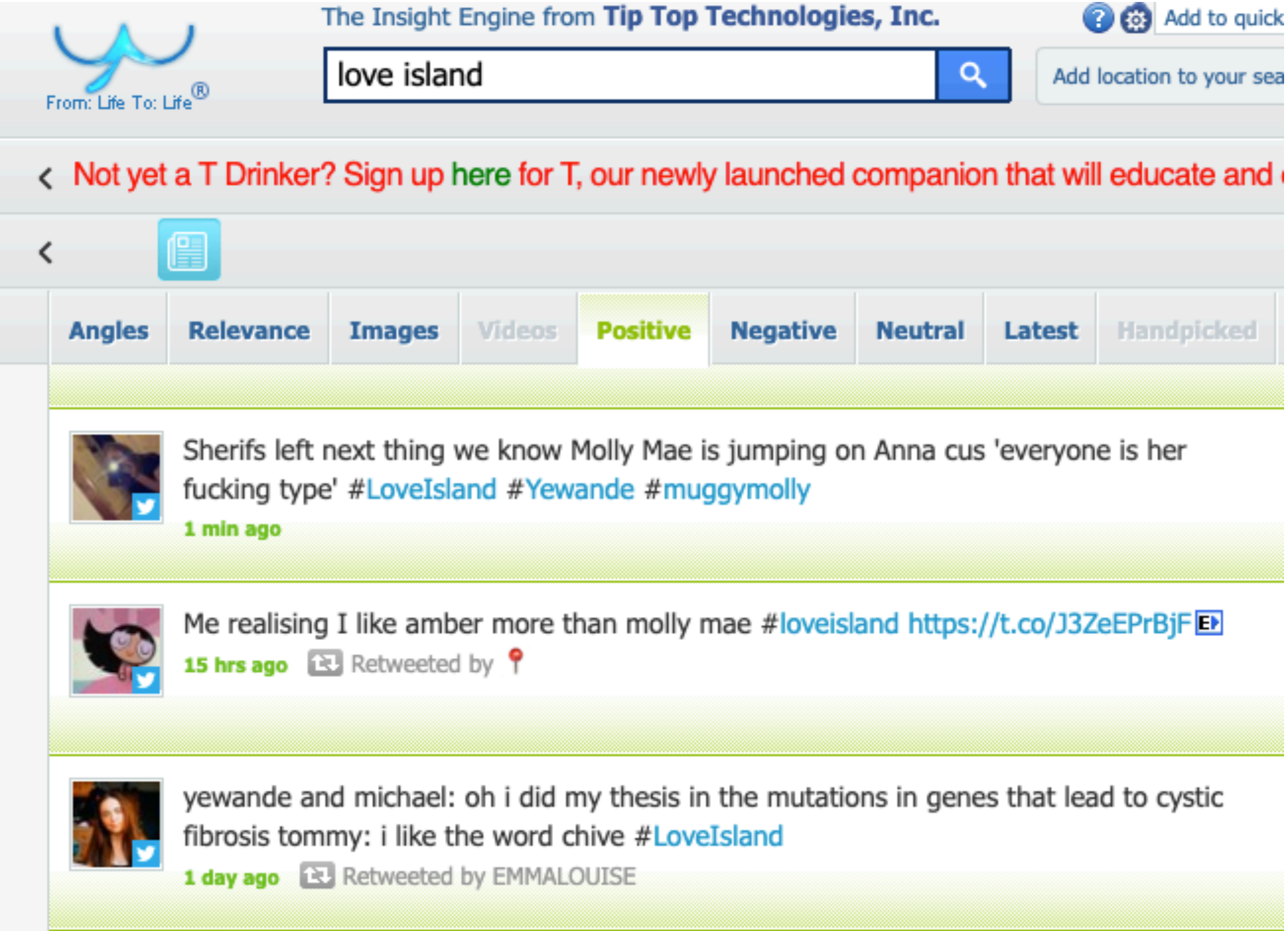
This document is: **negative (-0.090)**

reopens loss hoping frigid restore
tragic understanding impossible kind
trouble productive hopeful
shut down troubles inconvenience
eager scientific research damage
safety robust

Scroll down for full report

GATE General Architecture for Text Engineering

TipTop



ParallelDots

Try our free demo now by typing a sentence or choose from the options in the drop-down menu.

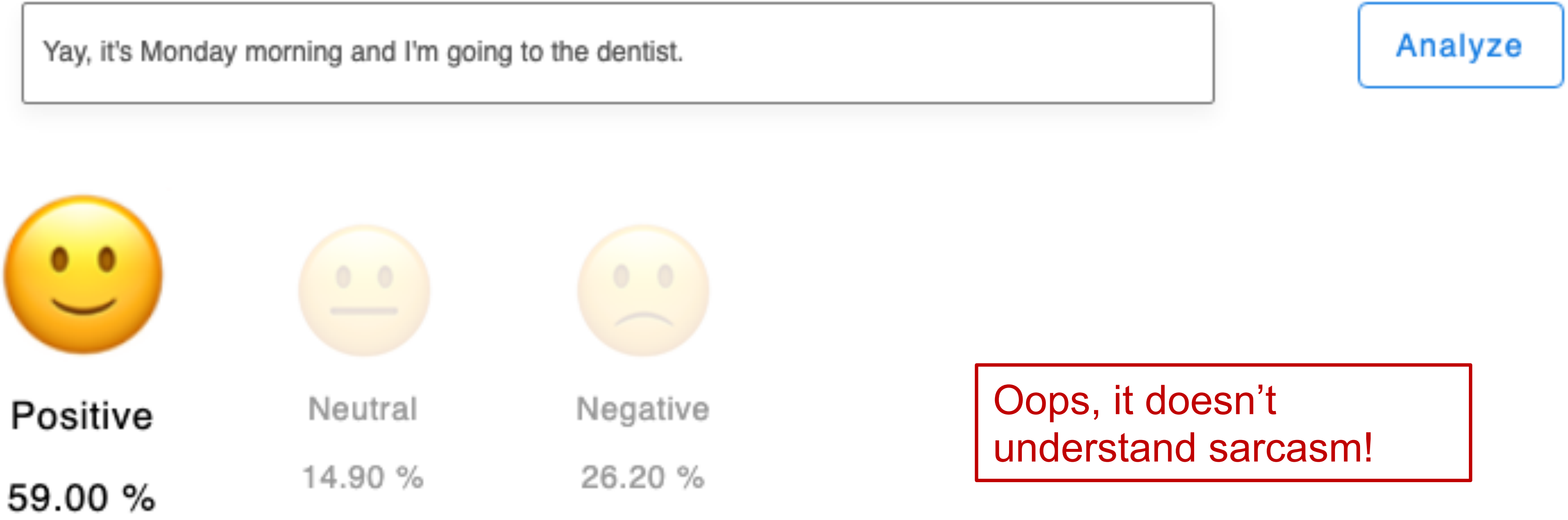
Select A Language English

It's Monday morning and I'm going to a funeral.

Analyze

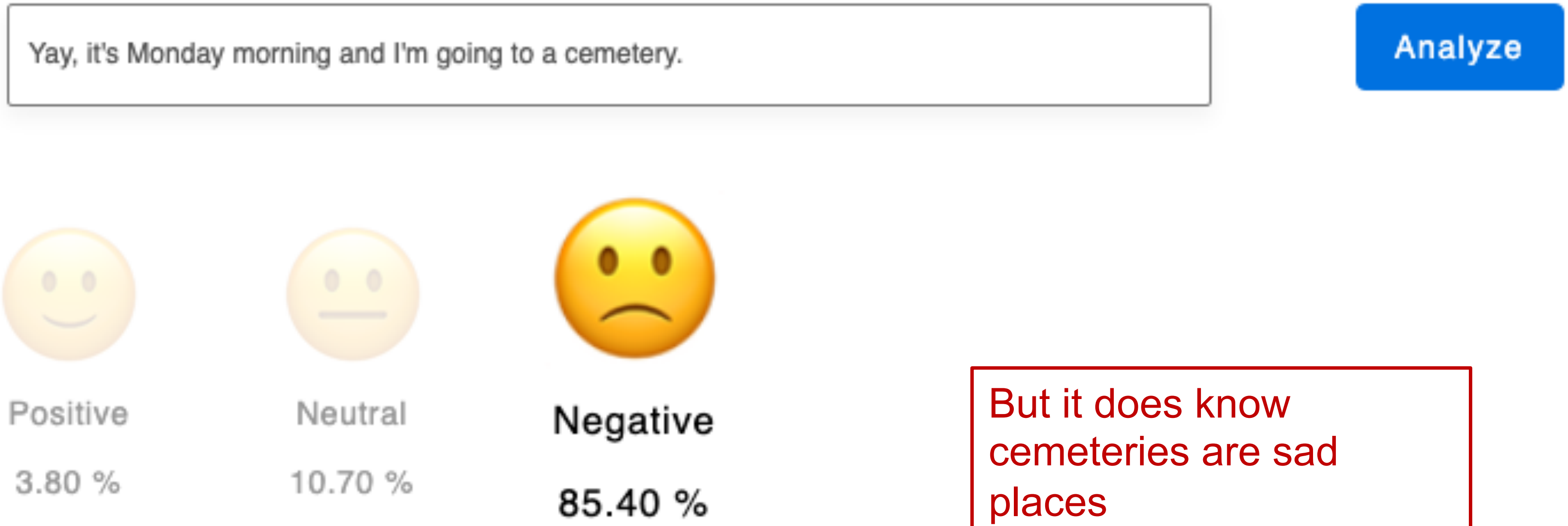
Positive	Neutral	Negative
0.60 %	7.10 %	92.30 %

ParallelDots (2)



Oops, it doesn't understand sarcasm!

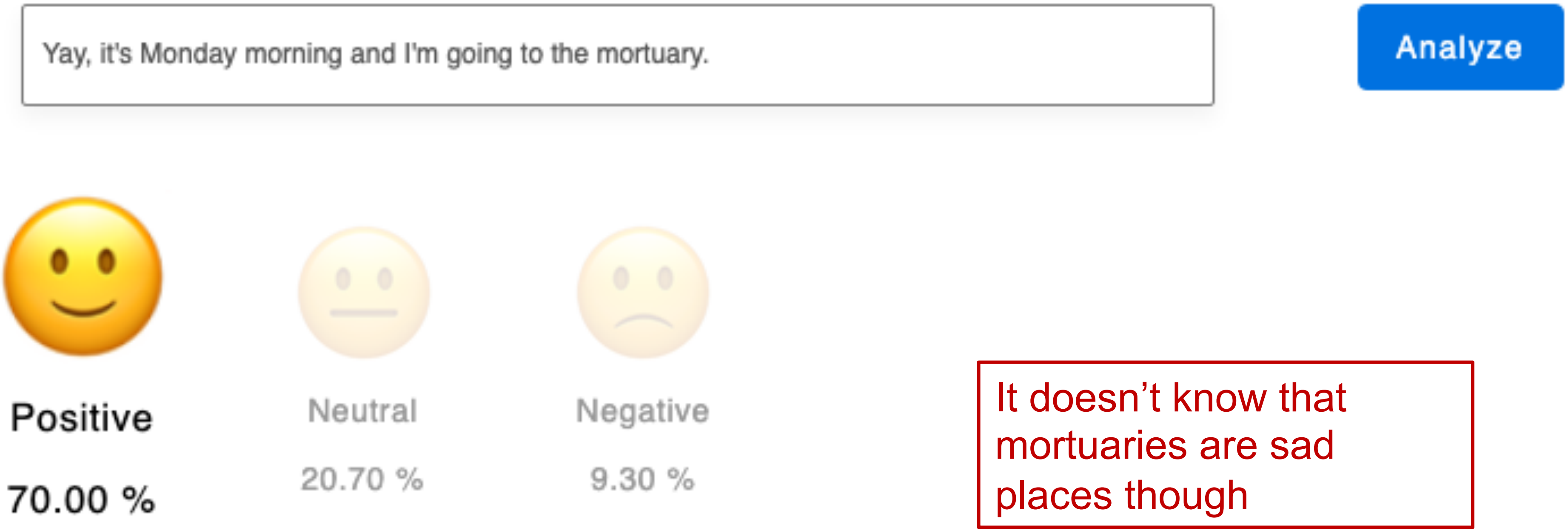
ParallelDots (3)



But it does know cemeteries are sad places

GATE General Architecture for Text Engineering

ParallelDots (4)



It doesn’t know that mortuaries are sad places though

NCSU Viz

Tweet Sentiment Visualization
Sentiment
Topics
Heatmap
Tag Cloud
Timeline
Map
Affinity
Narrative
Tweets
brexit (351)
pleasant
high confidence
unpleasant
low confidence
active
tense
alert
nervous
excited
stressed
elated
upset
happy
unpleasant
pleasant
sad
contented
unhappy
serene
depressed
relaxed
bored
calm
subdued
Keywords:
brexit
Query

Why not use these apps?

- Easy to search for opinions about famous people, brands and so on
- Hard to search for more abstract concepts, perform a non-keyword based string search
- E.g. many of the positive/negative tweets aren’t really about Love island, they’re about the characters in it
- They're suitable for a quick sanity check of social media, but not always for business needs
- Typically they need tailoring to your particular task/domain/application/data and they only exist as black box
- You can’t combine them with your own GATE annotations (we’ll see this in the abuse analysis module)

Why are they unsuccessful?

- Some don't work well at more than a very basic level
- They mainly use dictionary lookup for positive and negative words
- Tools based on supervised ML need similar text to training data
- Words appearing in different contexts might have different meanings
- They often don’t take account of aspect / opinion target - there is no correlation between the keyword and the sentiment: the sentiment refers to the tweet as a whole
- Sometimes this is fine, but it can also go horribly wrong

“Positive” tweets about fracking

- *Help me stop fracking. Sign the petition to David Cameron for a #frack-free UK now!*
- *I'll take it as a sign that the gods applaud my new anti-fracking country love song.*
- *#Cameron wants to change the law to allow #fracking under homes without permission. Tell him NO!!!!!*

Be careful!

Sentiment analysis isn't just about looking at the sentiment words

- *“It's a great movie if you have the taste and sensibilities of a 5-year-old boy.”*
- *“It's terrible that John did so well in the debate last night.”*
- *“I'd have liked the film a lot more if it had been a bit shorter.”*

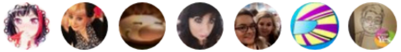
Situation is everything. If you and I are best friends, then my swearing at you might not be negative.

Swear words mean anger, right?

Please, everyone retweet this with me so they can see: Fuck Paddy Jackson, fuck Paddy Jackson, fuck Paddy Jackson get him off the team and the load of other rapists @IrishRugby NO PRIDE, NO PRIDE FOR YE, NOT MY IRELAND #IBelieveHer

5:59 AM - 28 Mar 2018

5 Retweets 4 Likes



Well, maybe not always

It really scares me when people wake up energized as fuck, like I'm literally immobile in the morning

6:06 AM - 28 Mar 2018

GATE General Architecture for Text Engineering

What about this?

If we specifically mention someone with a nasty word, that has to be bad, right?

You *$!%*&”!

Tell that to the victims of the Pulse massacre, you witch. You knew about that too! Sick!

8:54 AM - 26 Mar 2018

GATE General Architecture for Text Engineering

Well, not always….

Meanest thing I’ve done/said all week to @taylorc1103 was.
hands her a broom - careful. You might fly away. You witch 🙃
PS she luvs me it’s aight

10:10 AM - 26 Mar 2018

1 Retweet 1 Like 

Death confuses opinion mining tools

What did people think about Leonard Nimoy?

GATE General Architecture for Text Engineering

Public sentiment was overwhelmingly negative

Peter Cook @Slublog · May 22
Oh, hell. Watching the 2009 “Star Trek” just made me remember that **Leonard Nimoy** is **dead**. And now I'm sad.

3

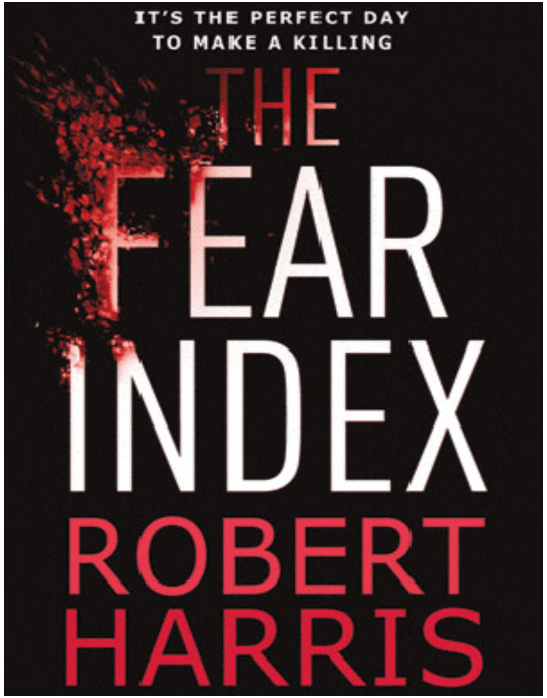
aahlejandro @Dank_Herbert · May 12
Has **Leonard Nimoy** been **dead** long enough that it's okay to say he wrote the worst poetry on the planet

Romantic Lost Prince @danaFringe · Mar 8
The great actor, **Leonard Nimoy** is **dead**, February 27 2015. I'm so very sad and I very much miss him! pinterest.com/pin/4321345267…

Usum @usumtwit · May 9
Leonard Nimoy Dead: 9 Great Moments of His Career Besides Spock ift.tt/1P8UyqH #News, **leonard**, **leonard nimoy**, **nimoy**, spock

Opinion Mining for Stock Market Prediction

- It might be only fiction, but using opinion mining for stock market prediction has been a reality for some years
- Research shows that opinion mining outperforms event-based classification for trend prediction [Bollen2011]
- Many investment companies offer products based on (shallow) opinion mining



Derwent Capital Markets

- Derwent Capital Markets launched a £25m fund that makes its investments by evaluating whether people are generally happy, sad, anxious or tired, because they believe it will predict whether the market will move up or down.
- Bollen told the Sunday Times: "*We recorded the sentiment of the online community, but we couldn't prove if it was correct. So we looked at the Dow Jones to see if there was a correlation. We believed that if the markets fell, then the mood of people on Twitter would fall*."
- "But we realised it was the other way round — that a drop in the mood or sentiment of the online community would precede a fall in the market."

GATE General Architecture for Text Engineering

Derwent Capital Markets

CAYMAN

" Using global sentiment analysis to trade the financial markets "

Social Media Sentiment Trading - Private Managed Accounts

Due to massive demand we have decided to apply our social media sentiment analysis technology to managed trading accounts allowing private investors the opportunity to invest upwards of £10,000 GBP.

If you are a sophisticated investor or high net worth person and not a US resident and would like to open an account then please click the link below...

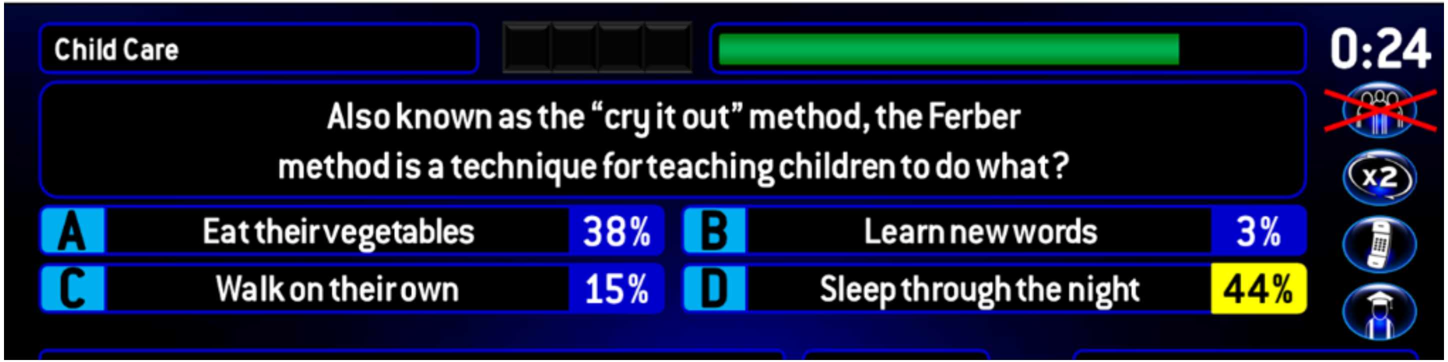
But don't believe all you read...

- It's not really possible to predict the stock market in this way
- Otherwise we'd be millionaires by now ☺
- In Bollen's case. the advertised results were biased by selection (they picked the winners after the race and tried to show correlation)
- The accuracy claim is too general to be useful (you can't predict individual stock prices, only the general trend)
- There's no real agreement about what's useful and what isn't
- http://sellthenews.tumblr.com/post/21067996377/noitdoesnot

Let’s play a game!

Unmute your microphone if you want to participate

Who Wants to be a Millionaire?



Ask the audience?

Or phone a friend?

Which do you think is better?

GATE General Architecture for Text Engineering

What's the capital of Spain?

A: Barcelona

B: Madrid

C: Valencia

D: Seville

What's the height of Mt Kilimanjaro?

A: 19,341 ft

B: 23,341 ft

C: 15,341 ft

D: 21,341 ft

Go for the majority or trust an expert?

- It depends what kind of question you're asking
- In Who Wants to Be a Millionaire, people tend to ask the audience fairly early on, because once the questions get hard, they can't rely on the audience getting it right

What's the height of Mt Kilimanjaro?

A: 19,341 ft
B: 23,341 ft
C: 15,341 ft
D: 21,341 ft

What's the capital of Spain?

A: Barcelona
B: Madrid
C: Valencia
D: Seville

Why bother with opinion mining?

- It depends what kind of information you want
- Don't use opinion mining tools to help you win money on quiz shows
- Recent research has shown that one knowledgeable analyst is better than gathering general public sentiment from lots of analysts and taking the majority opinion
  http://www.worldscinet.com/ijcpol/21/2104/S1793840608001949.html
- But only for some kinds of tasks

Whose opinion should you trust?

- Opinion mining gets difficult when the users are exposed to opinions from more than one analyst
- Intuitively, one would probably trust the opinion supported by the majority.
- But some research shows that the user is better off trusting the most credible analyst.
- Then the question becomes: who is the most credible analyst?
- Notions of trust, authority and influence are all related to opinion mining

GATE General Architecture for Text Engineering

All opinions are not equal

- Opinion Mining needs to take into account how much influence any single opinion is worth
- This could depend on a variety of factors, such as how much trust we have in a person's opinion, and even what sort of person they are
  - Need to account for:
  - experts vs non-experts
  - spammers
  - frequent vs infrequent posters
  - “experts” in one area may not be expert in another
  - how frequently do other people agree?

Trust Recommenders

Relationship (local) trust:

- If you and I both rate the same things, and our opinions on them match closely, we have high relationship trust.
- This can be extended to a social networking group --> web of trust.
- This can be used to form clusters of interests and likes/dislikes

Reputation trust:

- If you've recommended the same thing as other people, and usually your recommendation is close to what the majority of people think, then you're considered to be more of an expert and have high reputation trust.
- We can narrow reputation trust to opinions about similar topics

Related (sub)topics: general

- **Opinion extraction**: extract the piece of text which represents the opinion
  - *I just bought a new camera yesterday. It was a bit expensive, but the battery life is very good.*
- **Sentiment classification/orientation:** extract the polarity of the opinion (e.g. positive, negative, neutral, or classify on a numerical scale)
  - negative: *expensive*
  - positive: *good battery life*
- **Opinion summarisation**: summarise the overall opinion about something
  - price: negative, battery life: positive → overall 7/10

Feature-opinion association

- **Feature-opinion association**: given a text with target features and opinions extracted, decide which opinions comment on which features.
  - “*The battery life is good but not so keen on the picture quality*”
  - **Target identification**: which thing is the opinion referring to?
  - **Source identification**: who is holding the opinion?
  - There may be attachment and co-reference issues
  - “*The camera comes with a free case but I don't like the colour much.*”
  - Does this refer to the colour of the case or the camera?

General Architecture for Text Engineering

Getting the target right is crucial

10 of 120 people found the following review helpful:

I'll buy this book ..., March 15, 2010

By **T Boyer "seattleparent"** (Seattle) - See all my reviews

This review is from: The Big Short: Inside the Doomsday Machine (Hardcover)

the moment there is a 9.99 Kindle edition. I'll give it a four star rating just so I'm not drawn and quartered by the mob. (Though if you're buying a book based on average stars, without reading the reviews, well how much of a reader are you really?) I'm a big Michael Lewis fan, and I'm sorry his publisher is more interested in winning a pricing war with Amazon than with making the book available to E-book readers.

Help other customers find the most helpful reviews

Was this review helpful to you? Yes No

Report abuse | Permalink

Comments (14)

19 of 394 people found the following review helpful:

Kindle Users get The Big Short !!, March 15, 2010

By **JayRye** - See all my reviews

This review is from: The Big Short: Inside the Doomsday Machine (Hardcover)

Yes, we kindle users certainly got "The Big Short" on this title. It's really unfortunate. Kindle users take note, the Publisher is W.W. Norton and this decision to not publish a kindle version highlights that greed is not limited to the banking industry.

Help other customers find the most helpful reviews

Was this review helpful to you? Yes No

Report abuse | Permalink

Comments (14)

Opinion spamming

Not all reviews or opinions are “real”

Suppose we run a contest where people retweet our ad repeatedly, and the winner’s whoever loses the most followers.

Spam opinion detection (fake reviews)

- Sometimes people get encouraged or even paid to post “spam” opinions supporting a product, organisation, group, political party etc.
- An article in the New York Times discussed one such company who gave big discounts to post a 5-star review about the product on Amazon
  - http://www.nytimes.com/2012/01/27/technology/for-2-a-star-a-retailer-gets-5-star-reviews.html
  - Could be either positive or negative opinions
  - Generally, negative opinions are more damaging than positive ones
- We see this a lot on Twitter (e.g. Russian bots) – connections with misinformation (module 10)

How to detect fake opinions?

- Machine learning: train against known fakes
- Review content: lexical features, content and style inconsistencies from the same user, or similarities between different users
- Complex relationships between reviews, reviewers and products
- Publicly available information about posters (time posted, posting frequency etc)
- Detecting inconsistencies, contradictions, lack of entailment etc. is also relevant here

Opinion mining and social media

- Social media provides a wealth of information about a user's behaviour and interests:
  - *explicit*: John likes tennis, swimming and classical music
  - *implicit*: people who like skydiving tend to be big risk-takers
  - *associative*: people who buy Nike products also tend to buy Apple products
- While information about individuals isn't useful on its own, finding defined clusters of interests and opinions is
- If many people talk on social media sites about fears in airline security, life insurance companies might consider opportunities to sell a new service
- This kind of predictive analysis is all about understanding your potential audience at a much deeper level - this can lead to improved advertising techniques such as personalised ads to different groups

Social networks can trigger new events

- Not only can online social networks provide a snapshot of current or past situations, but they can actually trigger chains of reactions and events
- Ultimately these events might led to societal, political or administrative changes
- Since the Royal Wedding, Pilates classes became incredibly popular in the UK solely as a result of social media.
- Why?
- Pippa Middleton's bottom is the answer!
- Pictures of her bottom are allegedly worth more than those of her face!
- Viral events (e.g. ice bucket challenge), petitions etc.

Social media and politics

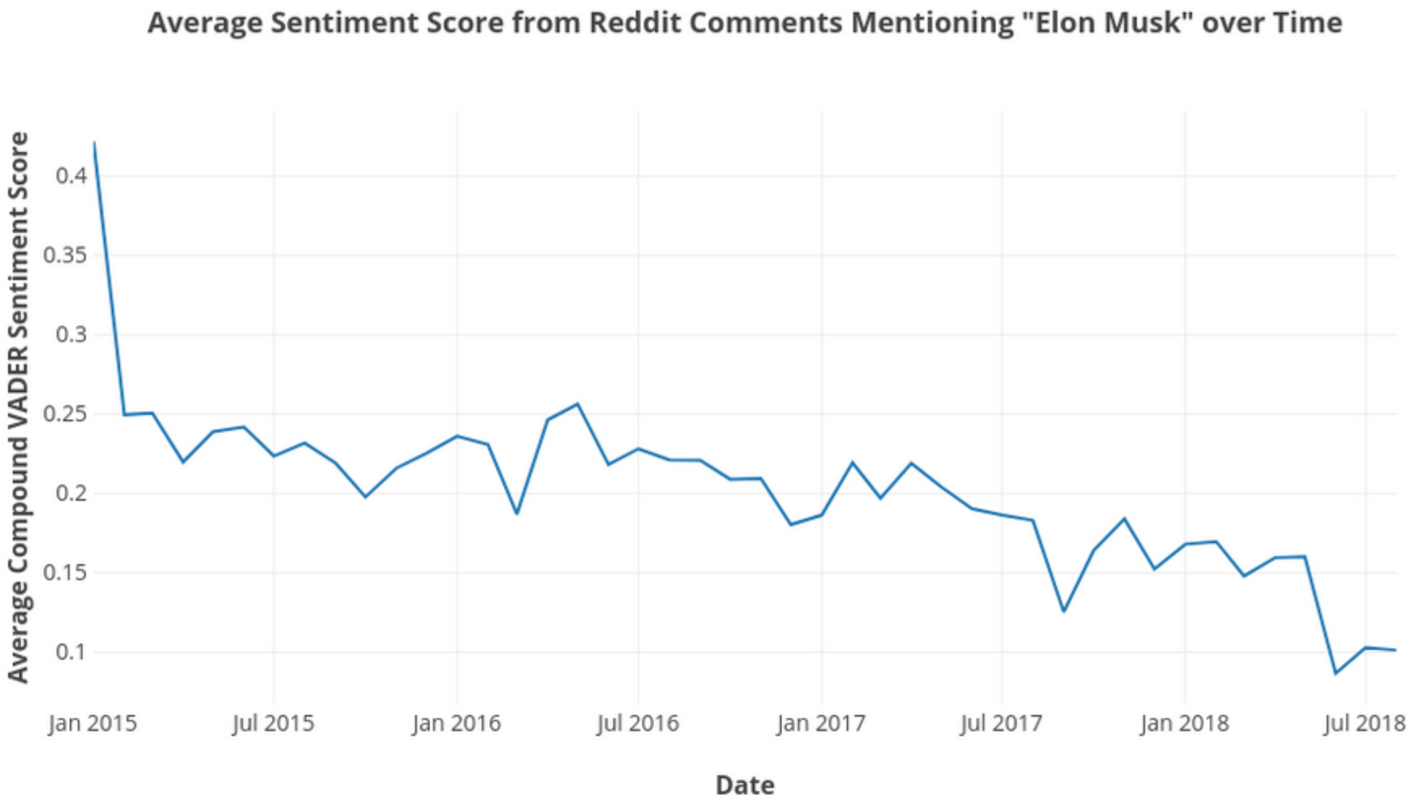
- Twitter provides real-time feedback on political debates that's much faster than traditional polling.
- Social media chatter can gauge how a candidate's message is being received or even warn of a popularity dive.
- Campaigns that closely monitor the Twittersphere have a better feel of voter sentiment, allowing candidates to fine-tune their message for a particular state: “playing to your audience".
- Examples of analysing tweets around UK elections and Brexit
  - http://services.gate.ac.uk/politics/ba-brexit
  - https://gate4ugc.blogspot.com/search/label/election%20tweet%20analysis
- Twitter has played a role in intelligence gathering on uprisings around the world, showing accuracy at gauging political sentiment, e.g. http://www.usatoday.com/tech/news/story/2012-03-05/social-super-tuesday-prediction/53374536/1

Tracking opinions over time

- Opinions can be extracted together with metadata such as time stamps and geo-locations
- We can then analyse changes to opinions about the same entity/event over time, and other statistics
- We can also measure the impact of an entity or event on the overall sentiment over the course of time

- We can also investigate correlations between events, topics, and time (see the Brexit study)

Viewing opinion changes over time



Some opinion mining resources

- Sentiment lexicons
- Sentiment-annotated corpora

General Architecture for Text Engineering

Bing Liu's English Sentiment Lexicon

- Available from http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar
- List of strings
- Positive words: 2006
    - a+, abound, abounds, abundance, abundant
- Negative words: 4783
    - Examples: 2-faced, abnormal, abolish, abominable
- Useful properties: includes mis-spellings, morphological variants, slang

The MPQA Subjectivity Lexicon

- GPL License; download from http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
  - type - either strongsubj or weaksubj
  - len - length of the clue in words
  - word1 - token or stem of the clue
  - pos1 - part of speech of the clue, may be anypos (any part of speech)
  - stemmed1 - y (yes) or n (no)
  - priorpolarity - positive, negative, both, neutral

	Strength	Length	Word	Part-of-speech	Stemmed	Polarity
1.	type=weaksubj	len=1	word1=abandoned	pos1=adj	stemmed1=n	priorpolarity=negative
2.	type=weaksubj	len=1	word1=abandonment	pos1=noun	stemmed1=n	priorpolarity=negative
3.	type=weaksubj	len=1	word1=abandon	pos1=verb	stemmed1=y	priorpolarity=negative
4.	type=strongsubj	len=1	word1=abase	pos1=verb	stemmed1=y	priorpolarity=negative
5.	type=strongsubj	len=1	word1=abasement	pos1=anypos	stemmed1=y	priorpolarity=negative
6.	type=strongsubj	len=1	word1=abash	pos1=verb	stemmed1=y	priorpolarity=negative

LIWC

- Linguistic Inquiry and Word Counts (LIWC) – a database of categorized regular expressions
- Costs around $90
- Some sentiment-relevant categories with example regular expressions below, from http://sentiment.christopherpotts.net/lexicons.html

Category	Examples
Negate	aint, ain't, arent, aren't, cannot, cant, can't, couldnt, ...
Swear	arse, arsehole*, arses, ass, asses, asshole*, bastard*, ...
Social	acquainta*, admit, admits, admitted, admitting, adult, adults, advice, advis*
Affect	abandon*, abuse*, abusi*, accept, accepta*, accepted, accepting, accepts, ache*
Anx	afraid, alarm*, anguish*, anxi*, apprehens*, asham*, aversi*, avoid*, awkward*
Anger	jealous*, jerk, jerked, jerks, kill*, liar*, lied, lies, lous*, ludicrous*, lying, mad

Problems with Sentiment Lexicons

Sentiment words are context-dependent and ambiguous

- a long dress” vs “a long walk” vs “a long battery life”
- “the camera was cheap” vs “the camera looked cheap”
- “I like her” vs “People like her should be shot”.

Solutions involve

- domain-specific lexicons
- lexicons including context (see e.g. Scharl's GWAP methods http://apps.facebook.com/sentiment-quiz)
- constraining POS categories

A general rule-based opinion mining application

Why Rule-based?

- Although ML applications are typically used for Opinion Mining, this task involves documents from many different text types, genres, languages and domains
- This is problematic for ML because it requires many applications trained on the different datasets, and methods to deal with acquisition of training material
- Aim of using a rule-based system is that the bulk of it can be used across different kinds of texts, with only the pre-processing and some sentiment dictionaries which are domain and language-specific

Application Stages

- Linguistic pre-processing
- Apply sentiment lexicons
- JAPE grammars (to do all the clever stuff)
- Aggregation of opinions

GATE General Architecture for Text Engineering

Linguistic pre-processing

- We first choose a pre-processing application such as TwitIE, ANNIE, or TermRaider
- Standard linguistic information (tokens, sentences etc.)
- Maybe language detection
- Named Entities or terms will provide us with information about possible opinion targets
- We could also do some topic or event recognition for the targets
- We can also choose not to have any specific targets

Basic approach for sentiment analysis

- Find sentiment-containing words in a linguistic relation with entities/events (opinion-target matching)
- Use a number of linguistic sub-components to deal with issues such as negatives, irony, swear words etc.
- Starting from basic sentiment lookup, we then adjust the scores and polarity of the opinions via these components

GATE General Architecture for Text Engineering

Sentiment finding components

- **Flexible Gazetteer Lookup**: matches lists of affect/emotion words against the text, in any morphological variant
- **Gazetteer Lookup:** matches lists of affect/emotion words against the text only in non-variant forms, i.e. exact string match (mainly the case for specific phrases, swear words, emoticons etc.)
- **Sentiment Grammars**: set of hand-crafted JAPE rules which annotate sentiments and link them with the relevant targets and opinion holders

Opinion scoring

- Sentiment gazetteers (developed from sentiment words in WordNet and other sources) have a starting “strength” score
- These get modified by context words, e.g. adverbs, swear words, negatives and so on

Some linguistic analysis

- We might want to consider some types of sentence differently
- Questions typically do not contain sentiment:
  - *“Do you think Sheffield has lovely weather?”*

is very different from

  - *“I think Sheffield has lovely weather.”*
- *Conditional sentences are also tricky:*
  - *“I wouldn’t live in Sheffield if I didn’t like the rain”.*
  - *”I would live in Sheffield if I liked the rain.”*
- So we might want to do some special pre-processing of sentence types

A (very) simple rule to find questions

```
Phase: Preprocess
Input: Token
Options: control = appelt

Rule: Question
(
 {Token.string == "?"}
):tag
-->
:tag.Question = {rule = "Question"}
```

Affect grammar

Make sure the POS is the same for the text and the gazetteer sentiment word.
Why?

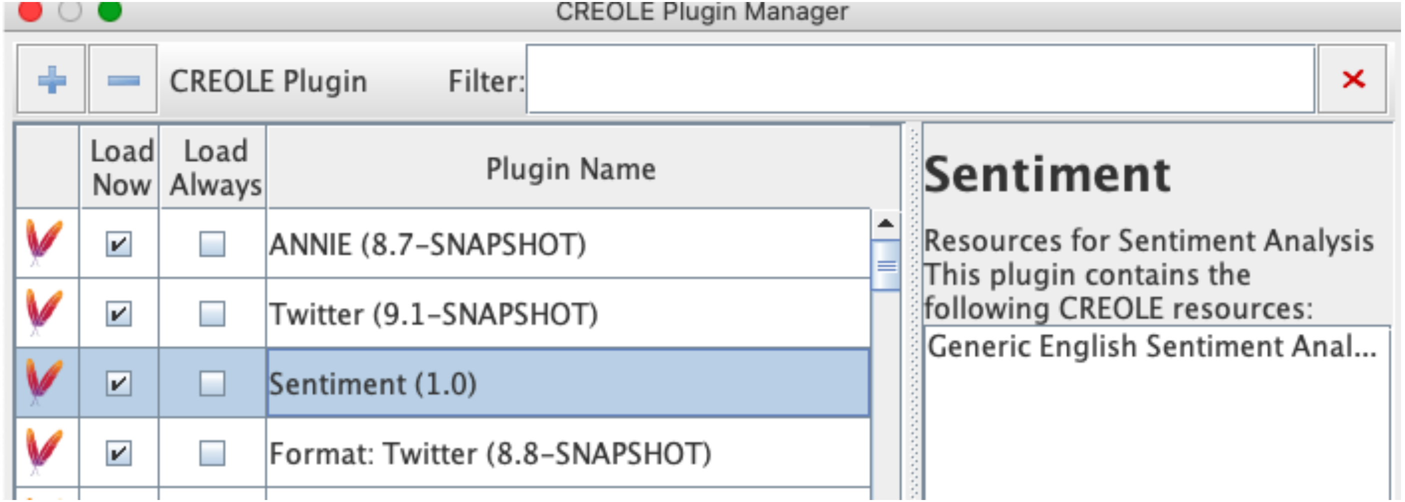
```
Phase:   Affect

Input: AffectLookup Token

Options: control = appelt


Rule: AffectAdjective

(

{AffectLookup.category == adjective,Token.category == VBN}|

{AffectLookup.category == adjective, Token.category == JJ}

):tag

-->

:tag.Affect = {kind = :tag.AffectLookup.kind,

      category = :tag.AffectLookup.category,  rule = "AffectAdjective"}
```

Check category of both Lookup and Token are adjectives or past participles

copy category and kind values from Lookup to new Affect annotation

Hands-on 1: GATE sentiment tool

- This tool depends on annotations produced by ANNIE, TwitIE or something else (these will be used as the opinion targets)
- In this exercise, we will use TwitIE
- Install the Sentiment ad Twitter plugins using the plugin manager



Hands-on 1: GATE sentiment tool (2)

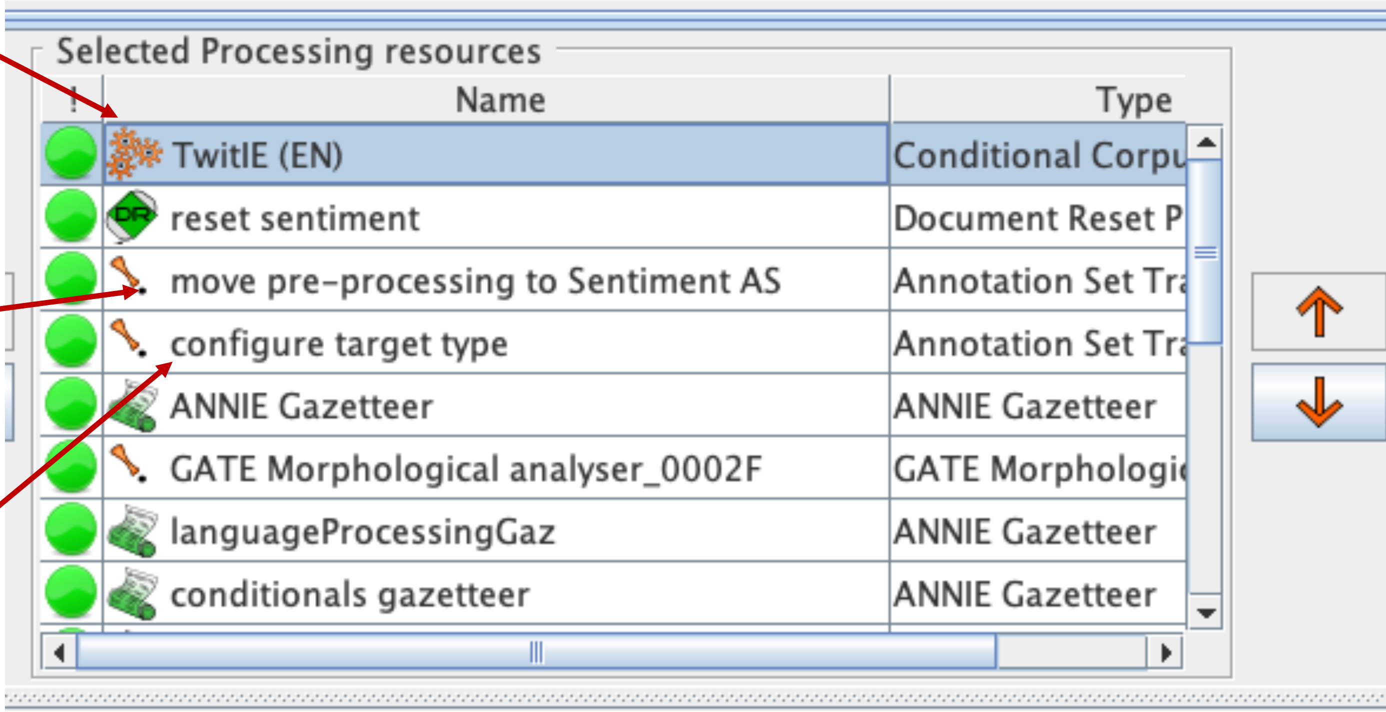
- Now in Applications -> Ready-made applications, you will find the Generic Sentiment Application under “Sentiment”
- Load this as well as the TwitIE application (from Ready-made applications -> TwitIE)
- Now we will combine the two applications together (you can add one Corpus Pipeline to another Corpus Pipeline)
- We do this by “pretending” that TwitIE is a PR
- Open the Sentiment application and add TwitIE from the set of Loaded Processing Resources on the left into the Sentiment application on the right
- Move TwitIE to be the first element (before the Document Reset)

Navigating the Sentiment application

We have put TwitIE "inside" the Sentiment Application

Here we move some annotations from TwitIE into the working annotation set (Sentiment)

Here we configure our opinion targets



GATE General Architecture for Text Engineering

The opinion target parameters

- “Configure target type” is an Annotation Set Transfer which allows you to decide what are the possible targets for your opinions.
- For example, these might be people, or specific people such as politicians, or any Named Entity, or anything else you want
- Because we’ve added TwitIE to our application, we can use any annotation type created by TwitIE as our target. By default, we use Person, Location and Organization

Corpus: <none>

Runtime Parameters for the "configure target type" Annotation Set Transfer:

Name	Type	Required	Value
annotationTypes	List		[Person=CandidateTarget, Location=CandidateTarget, Organization=CandidateTarget]
copyAnnotations	Boolean	✓	false
inputASName	String		Sentiment

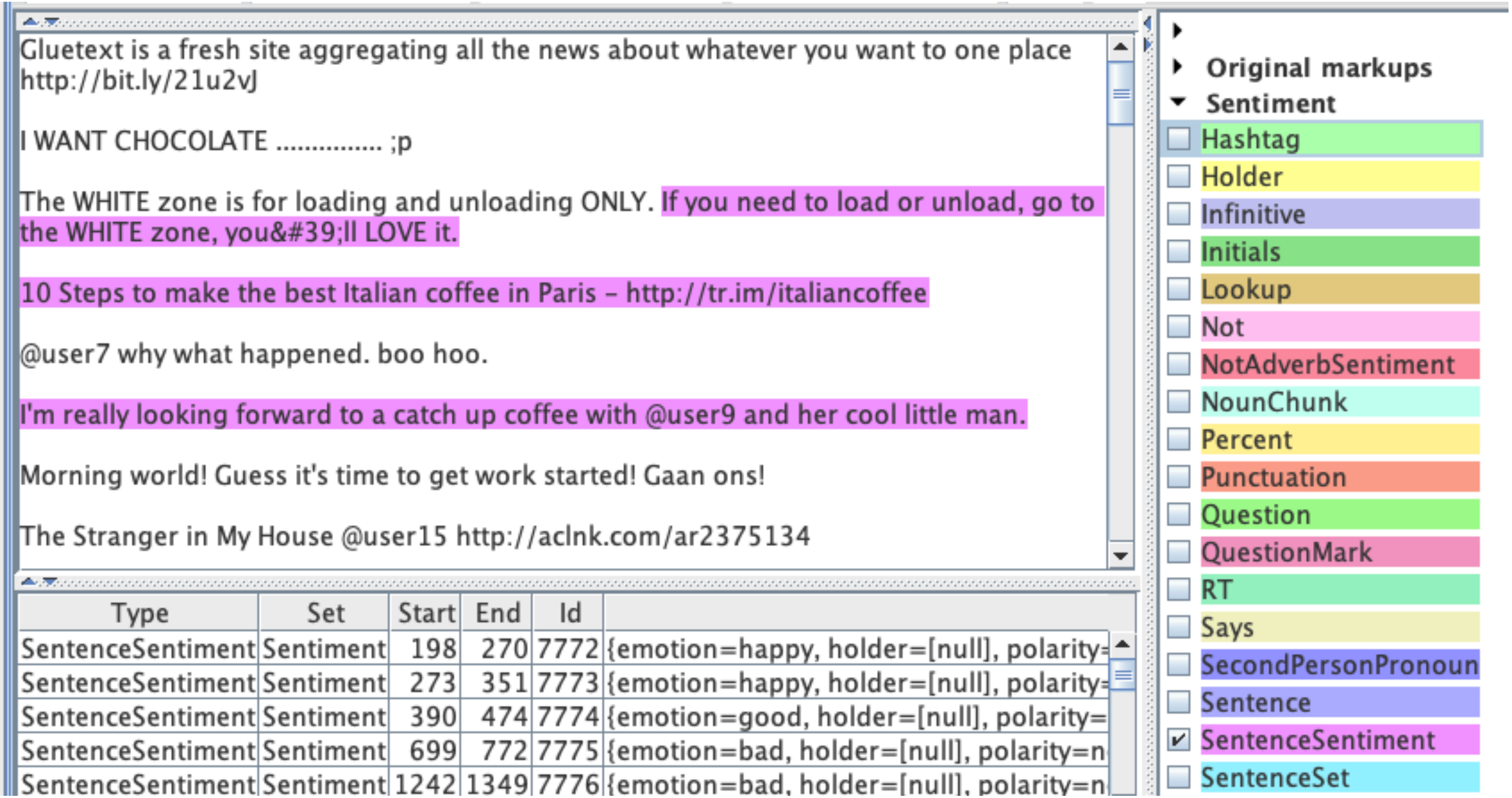
Run this Application

Running the GATE Sentiment tool

- Load the document *test-tweets-small.txt* and add it to a corpus
- Run the sentiment application on the document and check the results
- **The results are in the Sentiment annotation set**
- Hint: People often think they haven't done it right because they can't see the Sentiment in the Default set! Scroll down the annotation set pane until you see the Sentiment set.
- Each sentence containing a positive or negative sentiment is annotated with a SentenceSentiment annotation.
- Other annotations (e.g. Sentiment, SentimentTarget) give additional information

GATE General Architecture for Text Engineering

Your result should look something like this



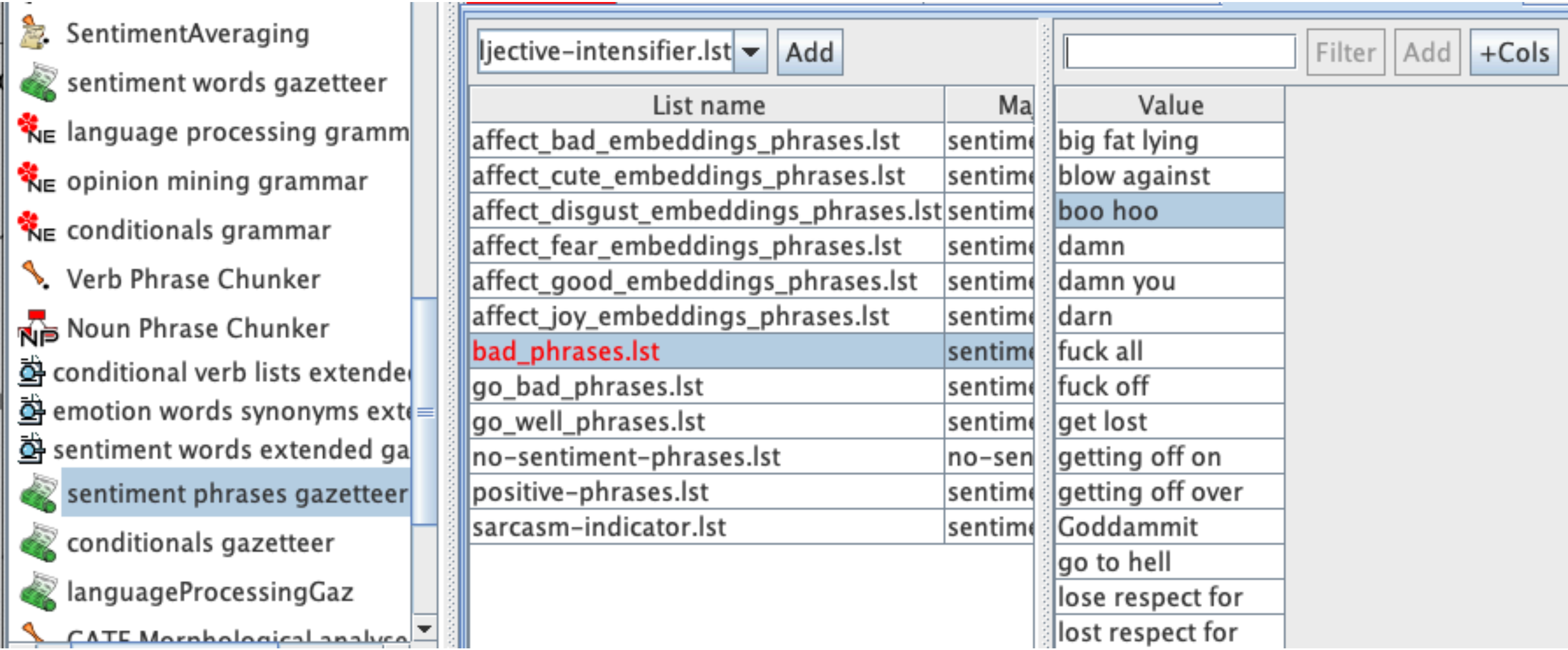
Let’s try improving it!

- Try modifying the gazetteers/grammars to improve the results
- A simple improvement might be to create a negative Sentiment annotation for the tweet “boo hoo”.
- To make changes to the application, first you need to make a copy of the sentiment plugin (download it via the plugin manager).
- Then unload the Sentiment application from GATE, load your saved version of it (from wherever you saved it) and repeat the process of adding TwitIE to it.
- A suggested solution is on the next page (but see if you can do it yourself first!)
- Hint: what would you need to change to get a new sentiment word to be recognised?

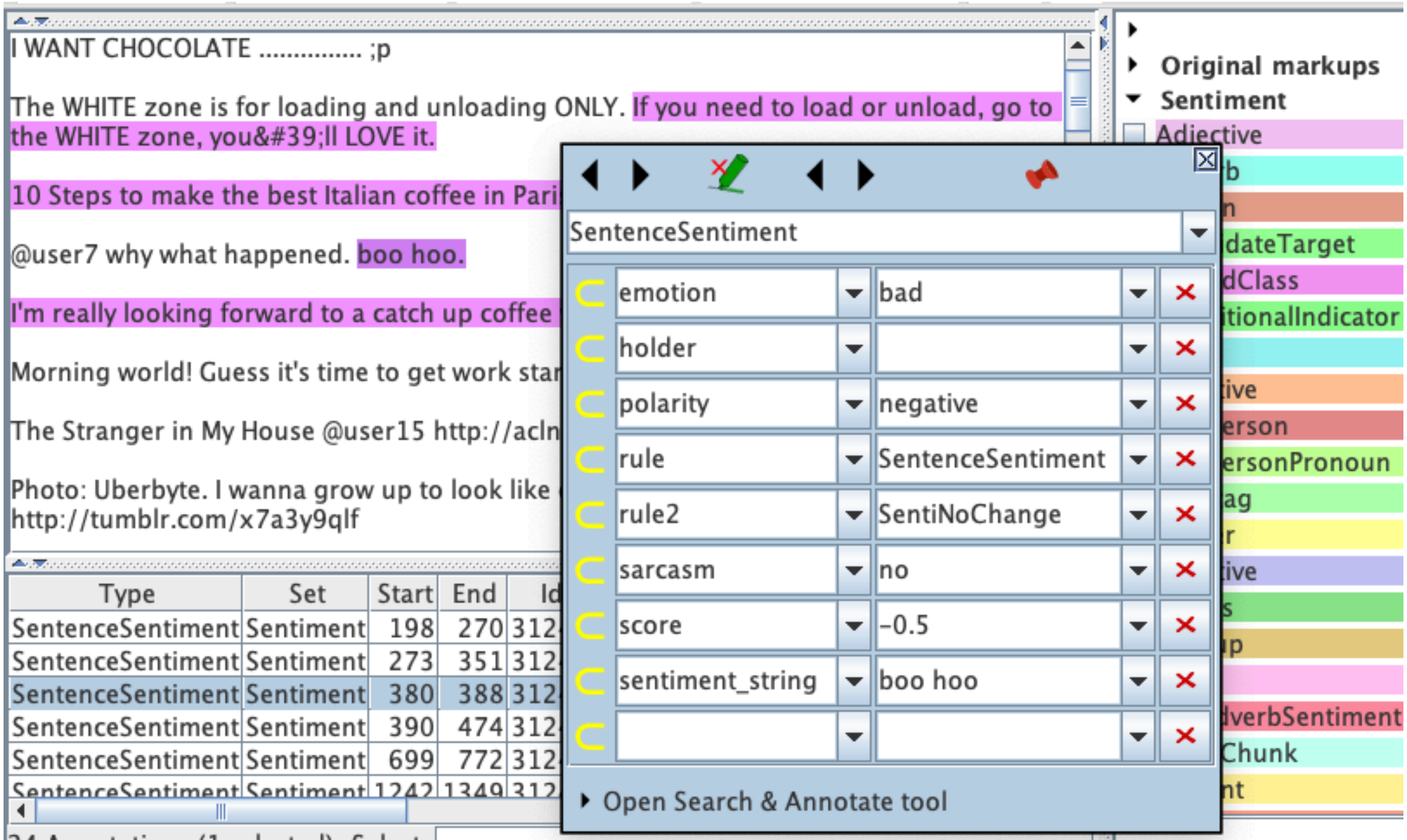
Hands-on 1: A possible solution

- Edit the list “bad-phrases.lst” in the sentiment phrases gazetteer by adding a new entry “boo hoo”
- Save and reinitialize the gazetteer
- Rerun the application
- Hint: there are many gazetteers in the application – double check that you have the right one (there are reasons for this)

Adding boo hoo to the gazetteer



If you did it right



Irony and sarcasm

- *I had never seen snow in Holland before but thanks to twitter and facebook I now know what it looks like. Thanks guys, awesome!*
- *Life's too short, so be sure to read as many articles about celebrity breakups as possible.*
- *I feel like there aren't enough singing competitions on TV . #sarcasmexplosion*
- *I wish I was cool enough to stalk my ex-boyfriend ! #sarcasm #bitchtweet*
- *On a bright note if downing gets injured we have Henderson to come in*

GATE General Architecture for Text Engineering

How do you tell if someone is being sarcastic?

- Use of hashtags in tweets such as #sarcasm, emoticons etc.
- Large collections of tweets based on hashtags can be used to make a training set for machine learning
- But you still have to know which bit of the tweet is the sarcastic bit

Man , I hate when I get those chain letters & I don't resend them , then I die the next day .. #Sarcasm

To the hospital #fun #sarcasm

What does sarcasm do to polarity?

- In general, when someone is being sarcastic, they're saying the opposite of what they mean
- So as long as you know which bit of the utterance is the sarcastic bit, you can simply reverse the polarity
- To get the polarity scope right, you need to investigate the hashtags: if there's more than one, you need to look at any sentiment contained in them.

Identifying the scope of sarcasm

- *I am **not happy** that I woke up at 5:15 this morning. #**great**start #sarcasm*
- negative sentiment + positive hashtag + sarcasm hashtag
- The positive hashtag becomes negative with sarcasm

- *You are **really mature**. **#lying** #sarcasm*
- positive sentiment + sarcasm indicator hashtag + sarcasm hashtag
- The positive sentiment is turned negative by both hashtags
- When in doubt, it's most likely that a sarcastic statement carries negative sentiment
- But there are also examples of “humble bragging”. Can you think of any?

What if you don't have a hashtag or other indicator?

- Look for word combinations with opposite polarity, e.g. “rain” or “delay” plus “brilliant”

Going to the dentist on my weekend home. Great. I'm totally pumped. #sarcasm

- Inclusion of world knowledge / ontologies can help (e.g. knowing that people typically don't like going to the dentist, or that people typically like weekends better than weekdays.
- It's an incredibly hard problem and an area where we expect not to get it right that often
- Most sarcasm detection is done using supervised ML, but people are often sarcastic in unusual ways

Machine Learning for Sentiment Analysis

Machine Learning for Sentiment Analysis

- ML is an effective way to classify opinionated texts
- We want to train a classifier to categorize free text according to the training data.
- Good examples are consumers' reviews of films, products, and suppliers.
- Sites like www.pricegrabber.co.uk show reviews and an overall rating for companies: these make good training and testing data
- We train the ML system on a set of reviews so it can learn good and bad reviews, and then test it on a new set of reviews to see how well it distinguishes between them
- We give an example of a real application and some related hands-on for you to try

Examples of consumer reviews

Merchant Info | **Merchant Ratings** | Uncategorized Products

Sort Reviews by: **Date** Rating

Write a Review »

Date Reviewed: 16/04/08

poet2000 **Member Since:** 16/04/08 **View Member's:** Reviews	30 days and still waiting	**Overall Rating**

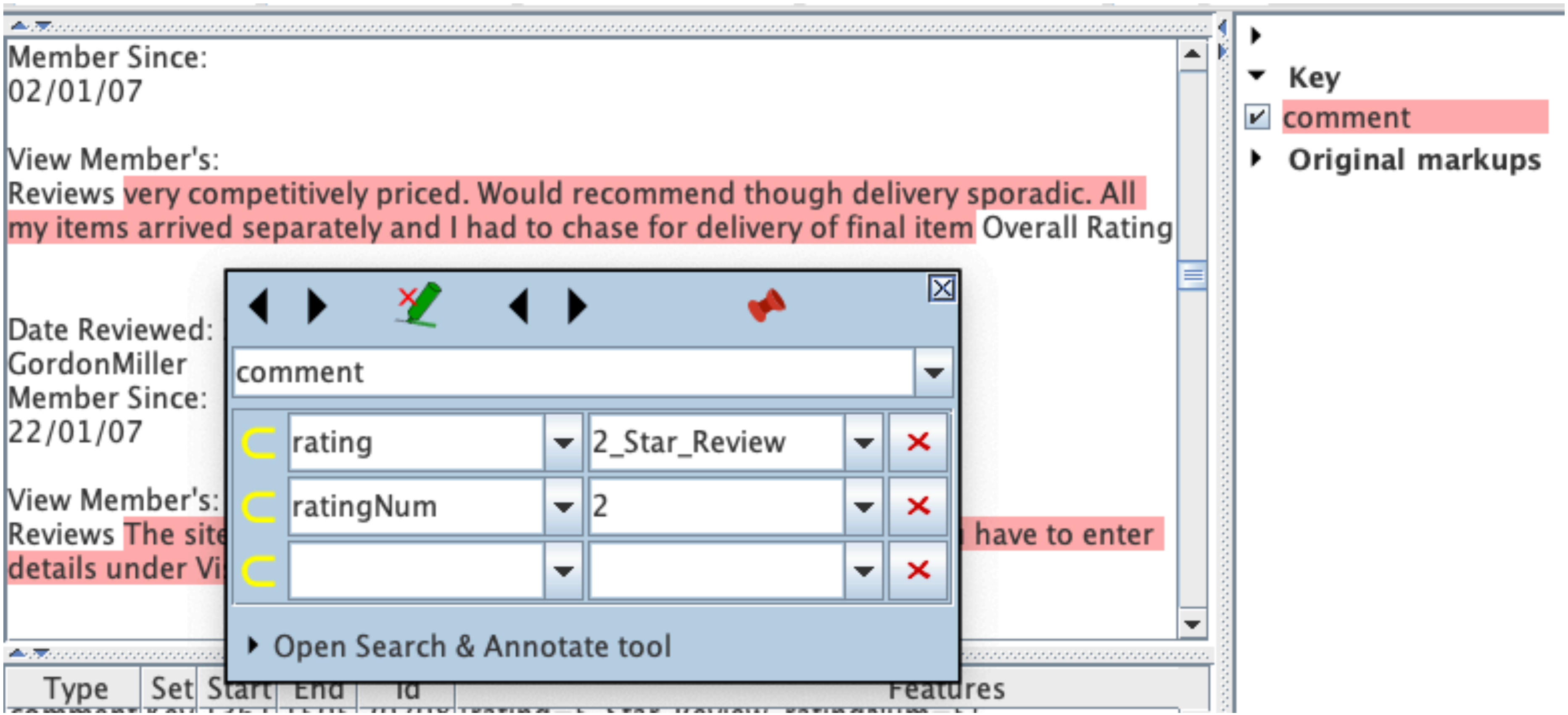
Date Reviewed: 24/01/07

Dbeach135 **Member Since:** 24/01/07 **View Member's:** Reviews	Jessops not only failed to complete the next day delivery, the item sent, a digital picture frame did not meet their specification. We ordered it as they claimed on their website that it accepted XD cards. This however was not the case. Jessops felt that they had done nothing wrong although their website was obviously wrong. This incorrect information still is outstanding and they have done nothing to correct their website even though I have notified them of the error.	**Overall Rating**

Preparing the corpus

- Corpus of 40 documents containing 552 company reviews.
- Each review has a 1- to 5-star rating.
- We pre-processed these in GATE to label each review with a comment annotation with a rating feature (free manual annotation!)
  - In ML terms:
  - instance = *comment* annotation
  - class = *rating* feature on the *comment* annotation
  - attributes = NLP features of the underlying text
- We will keep the spans of the comment annotations and use ML to classify them with the *rating* feature

Annotated review



Developing the training application

- We will develop an application that runs a set of NLP components to provide ML instance attributes, and trains the classifier
- Load the ANNIE, Tools, and Learning Framework plugins
- Create a new corpus called “training” and populate it from the directory **machine-learning/corpora/training** in the hands-on material
- Use a text editor to open the **machine-learning/feats.xml** config file so we can examine it

Feature Specification

```
<ML-CONFIG>
<NGRAM>
<NUMBER>1</NUMBER>
<TYPE>Token</TYPE>
<FEATURE>string</FEATURE>
</NGRAM>
</ML-CONFIG>
```

- The feature specification indicates we should use every Token string in the instance
- The “number” indicates to use unigrams

Building the training application (1)

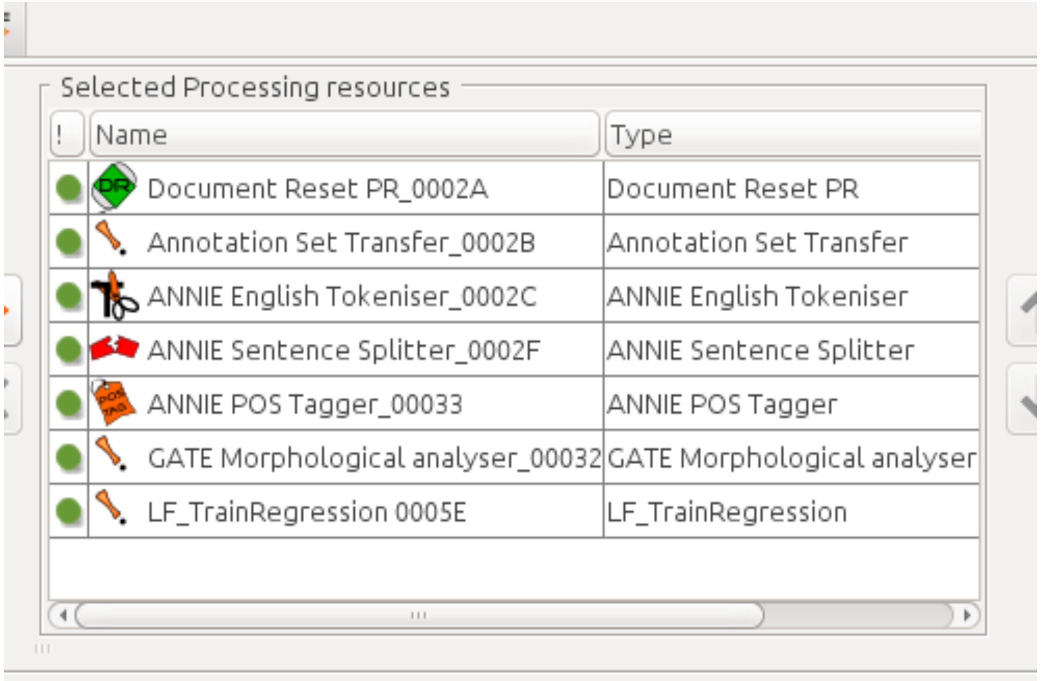
- Create the following PRs with the default init parameters:
  - Document Reset PR
  - Annotation Set Transfer
  - ANNIE English Tokeniser
  - ANNIE Sentence Splitter
  - ANNIE POS Tagger
  - GATE Morphological Analyser
  - LF_TrainRegression
  - LF_ApplyRegression
- Create a new Conditional Corpus Pipeline.

Building the application (2)

- We want to copy the comment annotations to the default annotation set to provide the ML instances and classes, but we don't want to remove the Key annotations
  - Add the following 2 PRs to the pipeline & set some runtime parameters:
  - Document Reset (default parameters)
  - Annotation Set Transfer:
    - annotationTypes = empty list (copy all)
    - copyAnnotations = true
    - inputASName = “Key”
    - outputASName & textTagName must be blank

Building the application (3)

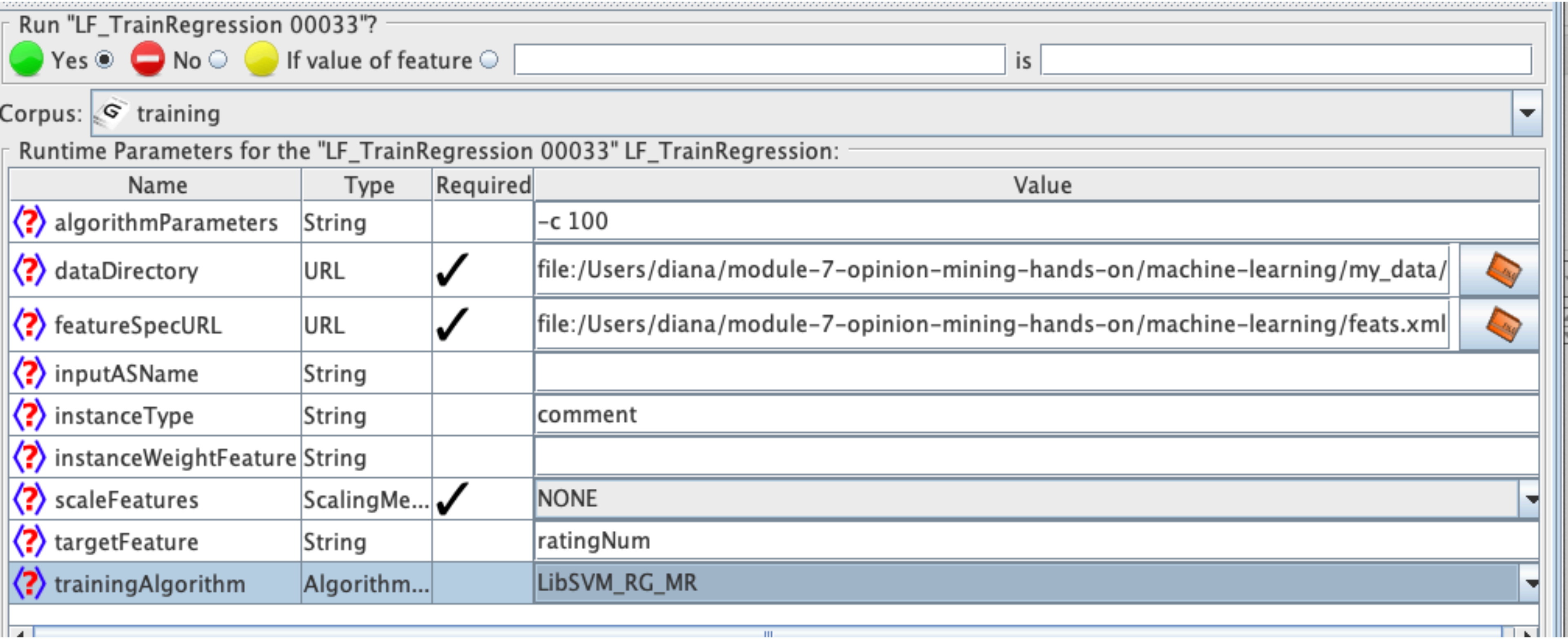
- Add the following loaded PRs to the end of your pipeline in this order:
  - English tokeniser
  - Sentence splitter
  - POS tagger
  - Morphological analyser
  - LF_TrainRegression



Learning Framework Parameters

- algorithmParameters: set to “-c 100” (explained in the ML module)
- dataDirectory is where the model will be saved. Create an empty directory and specify it here **(Remember this location – you will need it later!)**
- featureSpecURL is the feature specification file we inspected earlier (machine-learning/feats.xml)
- inputASName is the default annotation set (blank)
- instanceType is the name of the instance annotation type (“comment”)
- instanceWeightFeature & scaleFeatures can be ignored
- targetFeature is “ratingNum” (the numeric version)
- trainingAlgorithm is LIBSVM_RG_MR

Learning Framework Parameters



Algorithm and Target

- We are using a regression algorithm to do this task, because we are learning to predict numbers
- You could do this as a classification task by treating the ratings as words (using the "rating" feature), but numbers contain more information that words. We know that three is greater than one and less than five
- By using regression we can take into account that where the target is five, four is less wrong than one
- LIBSVM_RG uses a support vector machine to perform regression

Running the Training Application

- Run it on the training corpus (this should take less than 1 minute)
- The classifier's model is stored in the directory you indicated. The model is stored in text files, but they are not meant to be human-readable.
- In the Messages pane, you should see some messages, followed by:

LearningFramework: Training complete!

- Note that you won’t see anything interesting yet in your documents, as so far we have just trained our model on the training corpus, but we haven’t yet applied the model to our test data

Messages | Conditional Cor...

optimization finished, #iter = 735
nu = 0.5047458969513475
obj = -9391.701660002322, rho = 0.4024831499448613
nSV = 417, nBSV = 114
Engine class: class gate.plugin.learningframework.engines.EngineMBLibSVM
Algorithm: class gate.plugin.learningframework.engines.AlgorithmRegression
Trainer: [none]
Model: libsvm.svm_model@677f2e94
Info.engineClass: gate.plugin.learningframework.engines.EngineMBLibSVM
Info.algorithmClass: null
Info.algorithmParameters: -c 100
Info.task: REGRESSOR
Info.nrTrainingInstances: 453
Info.nrTrainingDocuments: 32
Info.nrTrainingDimensions: 2309
Info.nrTargetValues: 0
Info.classLabels: null
Info.trainingCorpus: training
Info.task: REGRESSOR
Info.seqEncoderClass: null
Info.seqEncoderOptions: null
Info.modelWhenTrained: 2021-02-17 15:26:02
Info.algorithmParameters: -c 100
FeatureInfo.growthStopped: false
FeatureInfo.globalScaling: NONE
FeatureInfo.feature: NgramAttribute(name=,type=Token,feature=string,featureName4Value=,number=1,maxlen=0,shorten=

LearningFramework: Training complete!

You should see something like this when training is complete. If not, check any error message and then check that you set all the parameters correctly, that the PRs are in the right order, and that you ran it on the right corpus.

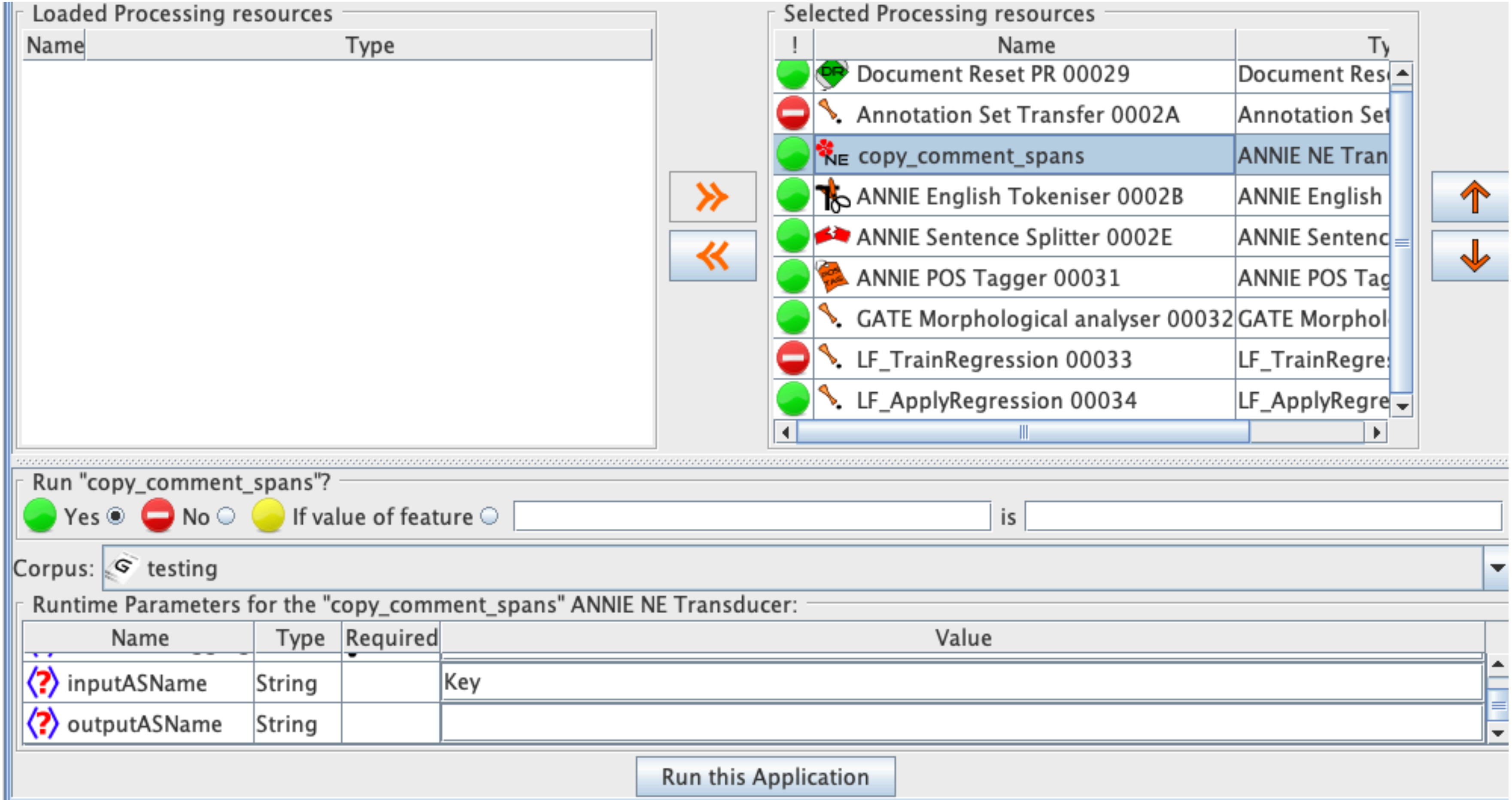
Applying the training model (1)

- Now we have trained our model, we can apply it to our test corpus to annotate our data.
- Create a “testing” corpus and populate it from the **corpora/testing** directory.
- (You may wish to remove your training corpus and its documents, to avoid confusion)
- To apply the model, we need to have comment annotations *without* rating features on the default AS. These will give us the instances to classify. A simple JAPE Transducer can do this.
- Create a JAPE PR using the grammar machine-learning/copy_comment_spans.jape
- Insert the grammar in the pipeline after the Annotation Set Transfer PR.
- Set the transducer parameters:
    - inputASName = “Key”
    - outputASName = “”

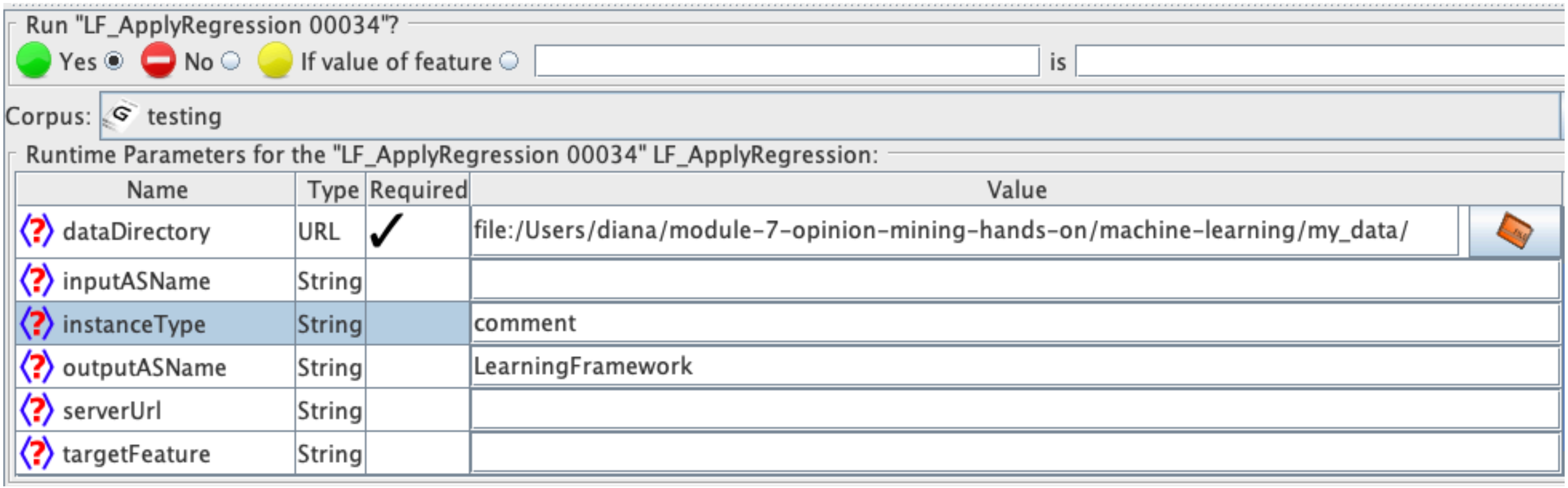
Applying the training model (2)

- Set the AS Transfer PR's run-mode to “no” (red)
- Set the LF_TrainRegression PR's run-mode to “no”
- Add the LF_ApplyRegression PR
- The classifier will get instances (*comment* annotations) and attributes (other annotations' features) from the default AS and put instances with classes (*rating* features) in the Output AS.

Your application should now look like this



LF_ApplyRegression Parameters



- dataDirectory is where you saved your model (same as the training PR's dataDirectory)
- instanceType is "comment" again
- outputASName defaults to LearningFramework, which helps us to clearly see what has been created
- targetFeature if left blank defaults to the same one as in training

Applying the training model (3)

Messages | Conditional Cor...

```
Starting application:
Engine class: class gate.plugin.learningframework.engines.EngineMBLibSVM
Algorithm: class gate.plugin.learningframework.engines.AlgorithmRegression
Trainer: [none]
Model: libsvm.svm_model@fcc1723
Info.engineClass: gate.plugin.learningframework.engines.EngineMBLibSVM
Info.algorithmClass: null
Info.algorithmParameters: -c 100
Info.task: REGRESSOR
Info.nrTrainingInstances: 453
Info.nrTrainingDocuments: 32
Info.nrTrainingDimensions: 2309
Info.nrTargetValues: 0
Info.classLabels: null
Info.trainingCorpus: training
Info.task: REGRESSOR
Info.seqEncoderClass: null
Info.seqEncoderOptions: null
Info.modelWhenTrained: 2021-02-17 15:26:02
Info.algorithmParameters: -c 100
FeatureInfo.growthStopped: false
FeatureInfo.globalScaling: NONE
FeatureInfo.feature: NgramAttribute(name=,type=Token,feature=string,featureName4Value=,number=1,maxlen=0,shorten=

Using target feature name from model: ratingNum
```

- Run the pipeline on the testing corpus
- When it's finished, you will see something like this

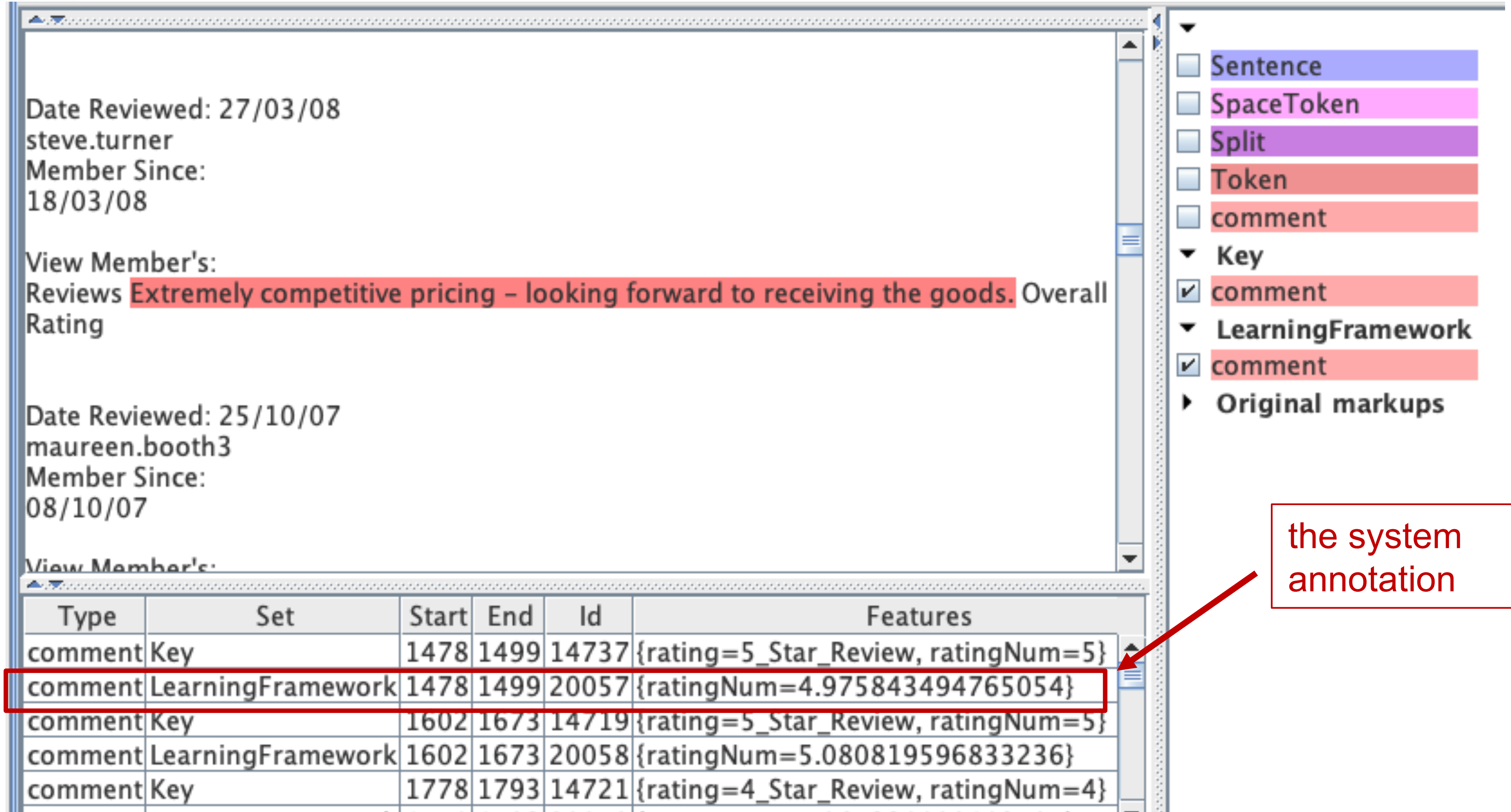
Inspecting the results

Open a few documents and inspect the "comment" annotations

The "answers" are in the LearningFramework set if you did it right:

- "Key" AS = user ratings (instances and correct classes)
- default AS = instances & attributes but no classes
- "LearningFramework" AS = instances with ratingNum values generated by ML

Annotation Results



Type	Set	Start	End	Id	Features
comment	Key	1478	1499	14737	{rating=5_Star_Review, ratingNum=5}
comment	LearningFramework	1478	1499	20057	{ratingNum=4.975843494765054}
comment	Key	1602	1673	14719	{rating=5_Star_Review, ratingNum=5}
comment	LearningFramework	1602	1673	20058	{ratingNum=5.080819596833236}
comment	Key	1778	1793	14721	{rating=4_Star_Review, ratingNum=4}

the system annotation

Evaluating the results

comment	Key	1478	1499	14737	{rating=5_Star_Review, ratingNum=5}
comment	LearningFramework	1478	1499	20057	{ratingNum=4.975843494765054}

- How can we compare the system (4.9758..) with the Key (5)?
- The values are real numbers, not integers, so Corpus QA will not work
- We can write a little JAPE grammar to convert the number back to a string
- Create a JAPE transducer PR from the numeric-to-string.jape file
- Add it to the end of the application and set both inputASName and outputASName to “LearningFramework”
- Run the application again: the output annotations now have additional “rating” features with values “1_Star_Review”, “2_Star_Review”, etc., so the results can be measured with Corpus QA and other tools

Cross-validation

- Cross-validation is a standard way to “stretch” the validity of a manually annotated corpus, because it enables you to test on a larger number of documents
- The 5-fold averaged result is more significant than the result obtained by training on 80% of the same corpus and testing on 20% once.

LF_EvaluateRegression

- The LF_EvaluateRegression PR will automatically split the corpus into 5 parts; then
  - train on parts 1,2,3,4; apply on part 5;
  - train on 1,2,3,5; apply on 4;
  - train on 1,2,4,5; apply on 3;
  - train on 1,3,4,5; apply on 2;
  - train on 2,3,4,5; apply on 1;
- and average the results. For regression, the PR will print the RMSE (root mean square error).

LF_EvaluateRegression

- To use the LF_EvaluateRegression PR, add it at the end of the pipeline and set the other PRs up as they were in the training mode (unused PRs can be switched off or removed).
- The evaluation PR uses the same runtime parameters as the training PR, plus a few for controlling cross-validation (or holdout).
- You can easily create a larger corpus from the all subdirectory (= training + testing).

Selected Processing resources

!	Name	Type
	Document Reset PR 00029	Document Reset PR
	Annotation Set Transfer 0002A	Annotation Set Transfer
	copy_comment_spans	ANNIE NE Transducer
	ANNIE English Tokeniser 0002B	ANNIE English Tokeniser
	ANNIE Sentence Splitter 0002E	ANNIE Sentence Splitter
	ANNIE POS Tagger 00031	ANNIE POS Tagger
	GATE Morphological analyser 00032	GATE Morphological analyser
	LF_TrainRegression 00033	LF_TrainRegression
	LF_ApplyRegression 00034	LF_ApplyRegression
	LF_EvaluateRegression 0008A	LF_EvaluateRegression

LF_EvaluateRegression Parameters

Run "LF_EvaluateRegression 0008A"?

Yes ◉ No ○ If value of feature ○ is

Corpus: all

Runtime Parameters for the "LF_EvaluateRegression 0008A" LF_EvaluateRegression:

Name	Type	Required	Value
algorithmParameters	String		-c 100
evaluationMethod	EvaluationMethod		CROSSVALIDATION
featureSpecURL	URL	✓	file:/Users/diana/module-7-opinion-mining-hands-on/machine-learning/feats.xml
inputASName	String		
instanceType	String		comment
instanceWeightFeature	String		
numberOfFolds	Integer		10
numberOfRepeats	Integer		1
scaleFeatures	ScalingMethod	✓	NONE
targetFeature	String		ratingNum
trainingAlgorithm	AlgorithmRegression		LibSVM_RG_MR
trainingFraction	Double		0.6667

GATE General Architecture for Text Engineering

Results

You should see something like this when finished

The result is displayed in terms of RMSE (the Root Mean Square Error)

Messages | Conditional Cor...

```
optimization finished, #iter = 549
nu = 0.476337522765944
obj = -8178.486903605694, rho = -0.814873625937015
nSV = 345, nBSV = 98
*
optimization finished, #iter = 575
nu = 0.45220839695621723
obj = -6835.2033152234435, rho = 1.9329189779831375
nSV = 346, nBSV = 89
*
optimization finished, #iter = 553
nu = 0.45910602188218025
obj = -7615.284941046269, rho = 1.3539653257644084
nSV = 347, nBSV = 90
*
optimization finished, #iter = 577
nu = 0.46133472522237723
obj = -7752.790388685204, rho = 0.19644763487387773
nSV = 352, nBSV = 93
Prob. model for test data: target value = predicted value + z,
z: Laplace distribution e^(-|z|/sigma)/(2sigma),sigma=0.443307515357011
*
optimization finished, #iter = 735
nu = 0.4713718178662083
obj = -9538.21208866647, rho = 1.07135070555620
nSV = 429, nBSV = 119
LearningFramework: Evaluation complete!
EvaluationResultRgXval{RMSE=0.7859408982665105,nrFolds=10}
```

GATE General Architecture for Text Engineering

The problem of sparse data

- One of the difficulties of drawing conclusions from traditional opinion mining techniques is the sparse data issue
- Opinions tend to be based on a very specific product or service, e.g. a particular model of camera, but don't necessarily hold for every model of that brand of camera, or for every product sold by the company
- One solution is figuring out which statements can be generalised to other models/products and which are specific
- Another solution is to leverage sentiment analysis from more generic expressions of motivation, behaviour, emotions and so on, e.g. what type of person buys what kind of camera?
- Contextual information is critical, but often this isn’t available

More information

- There are lots of papers about opinion mining on the GATE publications page https://gate.ac.uk/gate/doc/papers.html
- The EU-funded DecarboNet project dealt with monitoring sentiment about climate change in social media http://www.decarbonet.eu
- We used opinion mining to track sentiments by politicians on Twitter in the run-up to the UK 2015 and 2017 elections, in the Nesta-funded Political Futures Tracker project https://gate.ac.uk/projects/pft/
- We will also revisit opinion mining in the Online Abuse Detection module (module 9)

Extra Hands-on exercises

For the really brave

Hands-on 2: Using ANNIC with sentiment

- If you did module 6 (GATE Plugins) you will know how to use ANNIC
- Create a new Lucene datastore in GATE, using the default parameters, **but** set "AnnotationSets" parameter to exclude "Key" and "Original markups".
- Create a new empty corpus, save it to the datastore, then populate it with from the tweet-texts directory used from the social media module (module 4)
- Or use whatever dataset you want
- Close the corpus and documents in the viewer
- Double click on the datastore and double click on the corpus to load it
- Run the sentiment application (from the hands-on 1) on the corpus

Hands-on 2: Using ANNIC with sentiment

- Select “Lucene datastore searcher” from the datastore viewer
- Try out some patterns to see what results you get: if you find a pattern that enables you to find an opinion, try implementing it in a JAPE grammar
- Look for negative words in the tweets, and add some new gazetteer entries and/or grammar rules to deal with these.
- Look at Lookup, Token, Emoticon, Hashtag annotations in different combinations, for example
- You could always make up some new tweets and add them to the datastore, if you don't find examples of things like sarcasm or swearing
- NOTE: if you look at the documents individually, you may find some don’t have a Sentiment annotation set. This is because they don’t have any Entity annotations. If none of your documents have sentiment, you’ve done something wrong though!

Suggestions for further ML experiments...

Suggestions...

- The config file can be copied and edited with any text editor.
- Try n-grams where n>1
  - Change <NUMBER> in the config
  - Usually this is slower, but sometimes it improves quality
- Adjust the cost (-c value)
  - Increasing it may increase correct classifications, but can lead to overfitting.

Suggestions...

- Try using other features
- *Token.string*, *Token.category*, or combinations of these with *Token.root* and *Token.orth*
- You could even include other ANNIE PRs in the pipeline and use Lookup or other annotation types.
- You need to create the same attributes for training and application.
- If an instance does not contain at least one attribute (annotation+feature specified in the config file), the ML PR will throw a runtime exception, so it's a good idea to keep a Token.string unigram in the configuration.