Opinion Mining

What is Opinion Mining?

- OM is a recent discipline that studies the extraction of opinions using IR, AI and/or NLP techniques.
- More informally, it's about extracting the opinions or sentiments given in a piece of text
- Also referred to as Sentiment Analysis (though technically this is a more specific task)
- 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

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

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

But there are lots of tools that “analyse” social media already....

- Streamcrab http://www.streamcrab.com/
- Semantria http://semantria.com
- Social Mention http://socialmention.com/
- Sentiment140: http://www.sentiment140.com/
- TipTop: http://feeltiptop.com/

Why not use existing sentiment 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. to find opinions about Lady Gaga's dress, you can often only search on “Lady Gaga” to get hits
- They're suitable for a quick sanity check of social media, but not really for business needs
- And the opinion finding they do isn't always very good...

Why are these sites unsuccessful?

- They don't work well at more than a very basic level
- They mainly use dictionary lookup for positive and negative words
- Or they use ML, which only works for text that's similar in style
- They classify the tweets as positive or negative, but not with respect to the keyword you're searching for
- First, the keyword search just retrieves any tweet mentioning it, but not necessarily about it as a topic
- Second, 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 Candidate X 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.

Death confuses opinion mining tools

- Opinion mining tools are good for a general overview, but not for some situations

Nobody liked Leonard Nimoy

Or did they?

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

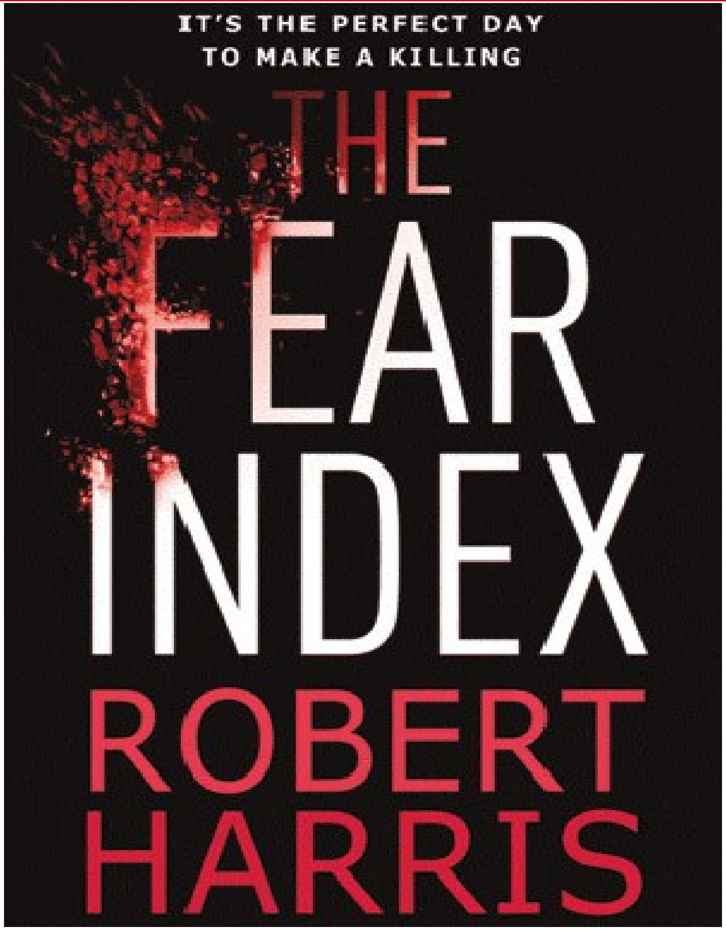
aahlejandro @Dank_Herbert · May 12
Has **Leonard Nimoy** been **dead** long enough that it's okay to say he wrote the worst poetry on the planet

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 already 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."

Derwent Capital Markets

CAYMAN

" Using global sentiment analysis to trade the financial markets "

Social Media Sentiment Trading - Private Managed Accounts

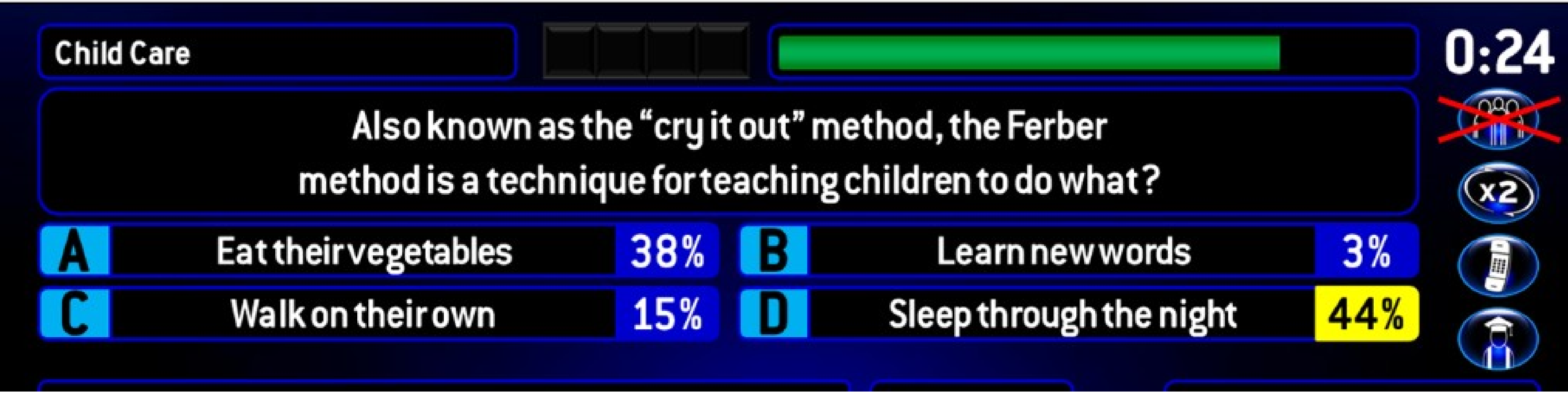
Due to massive demand we have decided to apply our social media sentiment analysis technology to managed trading accounts allowing private investors the opportunity to invest upwards of £10,000 GBP.

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

Who Wants to be a Millionaire?



Ask the audience?

Or phone a friend?

Which do you think is better?

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
- 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

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

- Two types of trust: relationship (local) trust and reputation (global) trust.
- **Relationship 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.
- **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 extend relationship trust to form clusters of interests and likes/dislikes
- 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. <u>It was a bit expensive, but the battery life is very good.</u>
- **Sentiment classification/orientation:** extract the polarity of the opinion (e.g. positive, negative, neutral, or classify on a numerical scale)
  - negative: <u>expensive</u>
  - positive: <u>good battery life</u>
- **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?

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

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 paid to post “spam” opinions supporting a product, organisation or even government
- 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?_r=3&ref=business
  - Could be either positive or negative opinions
  - Generally, negative opinions are more damaging than positive ones

How to detect fake opinions?

- Review content: lexical features, content and style inconsistencies from the same user, or simlarities 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!

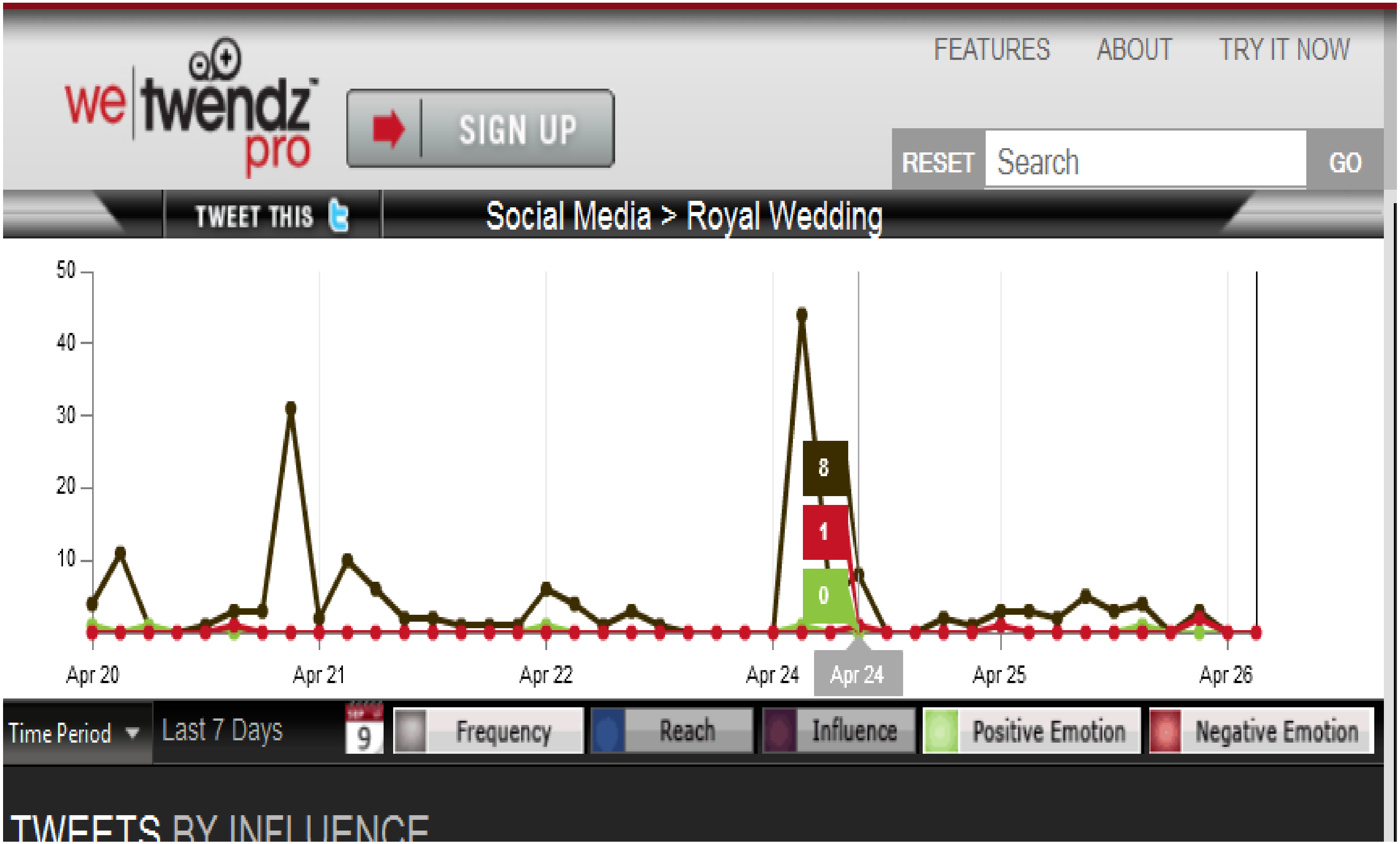
Social media and politics

- 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".
- But applying complex algorithms to social media is far from perfect for predicting politics, e.g. you can't detect sarcasm reliably.
- Nevertheless, Twitter has played a role in intelligence gathering on uprisings around the world, showing accuracy at gauging political sentiment.
- http://www.usatoday.com/tech/news/story/2012-03-05/social-super-tuesday-prediction/53374536/1

Tracking opinions over time

- Opinions can be extracted with a time stamp and/or a geo-location
- 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 about an entity or another event, over the course of time (e.g. in politics)
- Also possible to incorporate statistical (non-linguistic) techniques to investigate dynamics of opinions, e.g. find statistical correlations between interest in certain topics or entities/events and number/impact/influence of tweets etc.

Viewing opinion changes over time



Some opinion mining resources

- Sentiment lexicons
- Sentiment-annotated corpora

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

Rule-based Opinion Mining from Political Tweets

Processing political tweets

- Application to associate people with their political leanings, based on pre-election tweets
  - First stage is to find triple <Person, Opinion, Political Party>
  - e.g. John Smith is pro_Labour
  - Usually, we will only get a single sentiment per tweet
- Later, we can collect all mentions of “John Smith” that refer to the same person, and collate the information
  - For example, John may be equally in favour of several different parties, not just Labour, but hates the Conservatives above all else

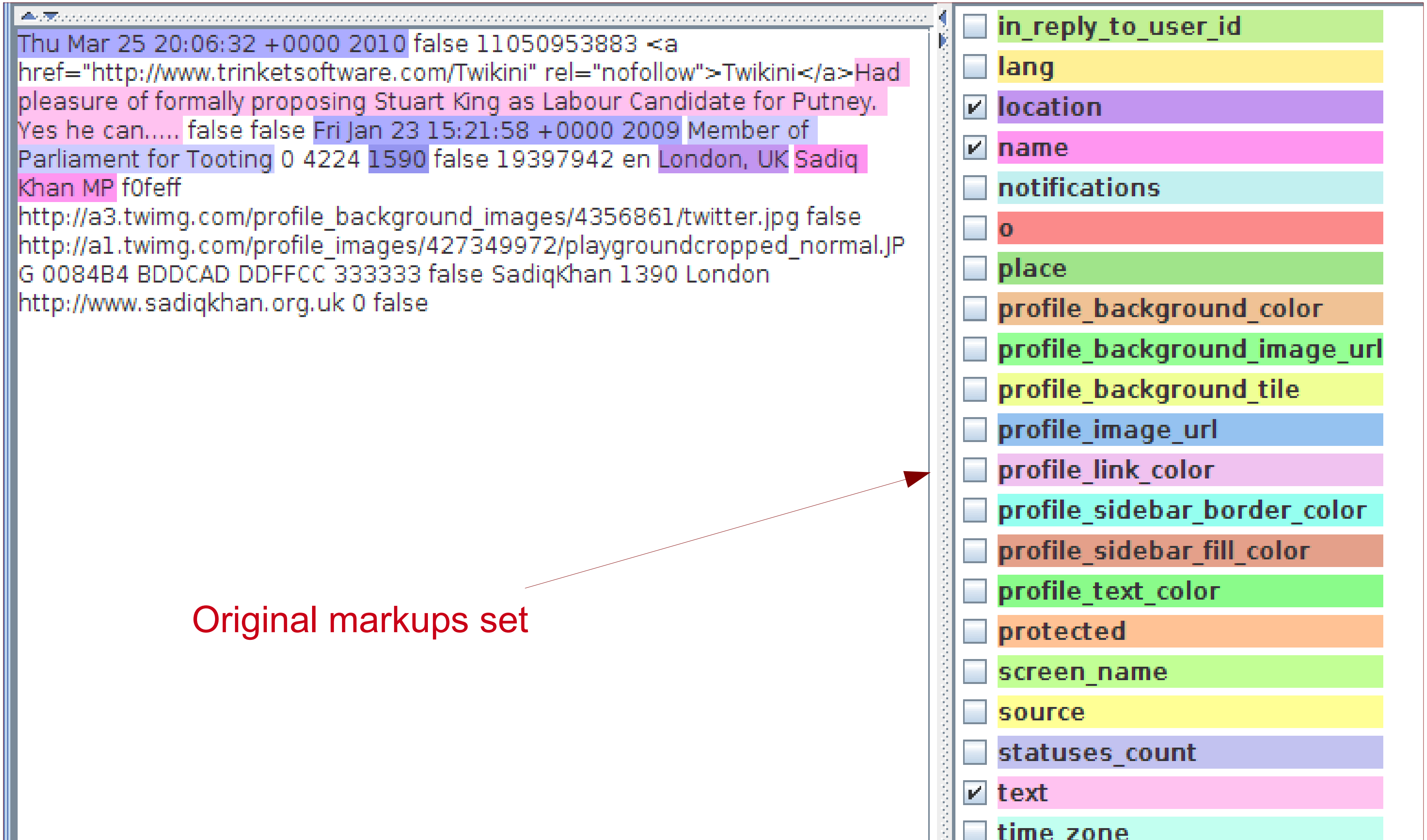
Creating a corpus

- First step is to create a corpus of tweets
- Used the Twitter Streaming API to suck up all the tweets over the pre-election period according to various criteria (e.g. use of certain hash tags, mention of various political parties etc.)
- Collected tweets in JSON format and then converted these to xml using a Python JSON library (now we have JSON import though)
- This gives us lots of additional twitter metadata, such as the date and time of the tweet, the number of followers of the person tweeting, the location and other information about the person tweeting, and so on
- This information is useful for disambiguation and for collating the information later

Corpus Size

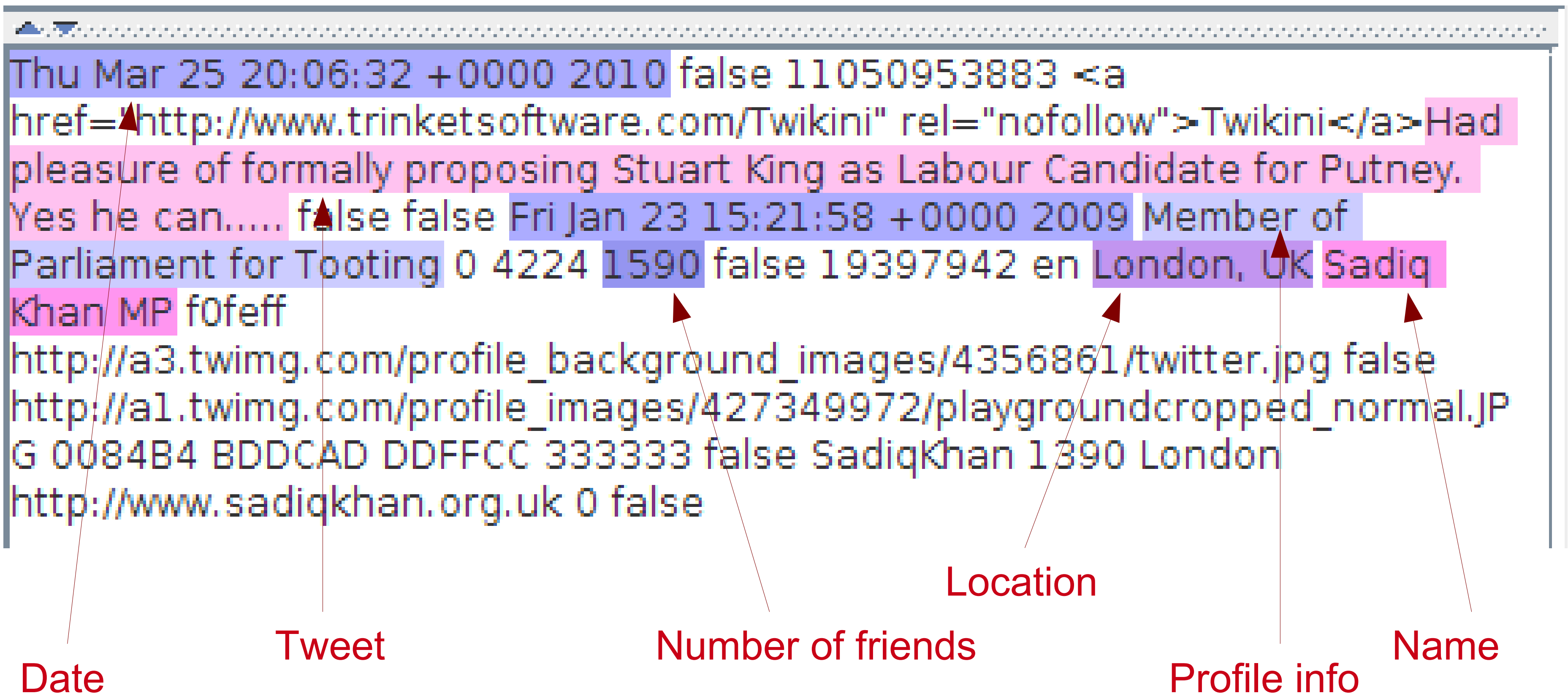
- Raw corpus contained around 5 million tweets
- Many were duplicates due to the way in which the tweets were collected
- Added a de-duplication step during the conversion of json to xml
- This reduced corpus size by 20% to around 4 million
- This still retains the retweets, however

Tweets with metadata



Original markups set

Metadata


Thu Mar 25 20:06:32 +0000 2010 false 11050953883 <a
href="http://www.trinketsoftware.com/Twikini" rel="nofollow">TwikiniHad
pleasure of formally proposing Stuart King as Labour Candidate for Putney.
Yes he can..... false false Fri Jan 23 15:21:58 +0000 2009 Member of
Parliament for Tooting 0 4224 1590 false 19397942 en London, UK Sadiq
Khan MP f0feff
http://a3.twimg.com/profile_background_images/4356861/twitter.jpg false
http://a1.twimg.com/profile_images/427349972/playgroundcropped_normal.JP
G 0084B4 BDDCAD DDFFCC 333333 false SadiqKhan 1390 London
http://www.sadiqkhan.org.uk 0 false
Location
Tweet
Number of friends
Name
Date
Profile info

Gazetteers

- We create an instance of a flexible gazetteer to match certain useful keywords, in various morphological forms:
  - political parties, e.g. “Conservative”, “LibDem”
  - concepts about winning election, e.g. “win”, “landslide”
  - words for politicians, e.g. “candidate”, “MP”
  - words for voting and supporting a party/ person, e.g. “vote”
  - words indicating negation, e.g. “not”, “never”
- We create another gazetteer containing affect/emotion words from WordNet.
  - these have a feature denoting part of speech (category)
  - Keeping category information may be important, so we don't want a flexible gazetteer here

Grammar rules: creating temporary annotations

- Identify questions or doubtful statements as opposed to "factual" statements in tweets.
    - Initially, we just look for question marks
    - “*Wont Unite's victory be beneficial to Labour?*”

- Create temporary Affect annotations if an “affect” Lookup is found and if the category matches the POS tag on the Token (this ensures disambiguation of the different possible categories)
    - *“Just watched video about <u>awful</u> days of Tory rule” vs “Ah <u>good</u>, the entertainment is here.”*
    - *“People <u>like</u> her should be shot.” vs “People <u>like</u> her.”*

Question grammar

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


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

Affect grammar

```
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

Grammar rules: finding triples

- We first create temporary annotations for Person, Organization, Vote, Party, Negatives etc. based on gazetteer lookup, NEs etc.
  - We then have a set of rules to combine these into pairs or triples:
  - *<Person, Vote, Party> "Tory Phip admits he voted LibDem".*
  - *<Party, Affect> "When they get a Tory government they'll be sorry."*
- We create an annotation "Sentiment" which has the following features:
  - kind = "pro_Labour", "anti_LibDem", etc.
  - opinion_holder = "John Smith", "author" etc.

Identifying the Opinion Holder

- If the opinion holder in the pattern matched is a Person or Organization, we just get the string as the value of opinion_holder
- If the opinion holder in the pattern matched is a pronoun, we first find the value of the string of the antecedent and use this as the value of opinion_holder
- Currently we only match opinion holders within the same sentence.
- If no explicit opinion holder then we use "author" as the value of opinion_holder.
- We could also use the metadata about the author of the tweet

Grammar rules: finding antecedents

- Find the antecedents of pronouns within a sentence so that we can refer a sentiment back to the original opinion holder or object of the opinion.
- First run the pronominal coreference PR
- Then use a JAPE rule to find pronouns linked to a Person or Organization
- We can identify these because they will have the feature “ENTITY_MENTION_TYPE” (created by the coreferencer)
- The co-referring pronouns all have also an antecedent_offset feature pointing to the proper noun antecedent
- The matching proper noun antecedent is found and its string is added as a feature on the relevant pronoun annotation

Creating the Application

- We only want to process the actual text of the tweet, not all the other information
- To do this, we use a Segment Processing PR to run the sentiment app over just the "text" annotation in Original Markups set.
- So, we need two applications: one containing the Segment Processing PR and one containing the actual sentiment application

Runtime Parameters for the "Segment Processing PR_0001E" Segment Processing PR:

Name	Type	Required	Value
analyser	LanguageAnalyser	✓	sentiment processing app
inputASName	String		Original markups
segmentAnnotationFeatureName	String		
segmentAnnotationFeatureValue	String		
segmentAnnotationType	String	✓	text

Linguistic information for better analysis

- Linguistic information can give you a lot of clues about meaning
- “Good battery life” seems to indicate a positive feature.
- But conditional sentences can have subtly different meanings:
- *I'd have bought a Nikon if I'd wanted good battery life*
- *I'll buy a Nikon if it has good battery life*
- *I'll buy a Nikon if I want good battery life*
- *I'd buy a Nikon unless I wanted good battery life*
- *I'd buy a Nikon even if it doesn't have good battery life.*

Conditional Types

0. If a camera has 20 hours of battery life, you can take many pictures.

- statement of fact or certainty

1. If someone makes a camera with 20 hours of battery life, I'll buy it

- potential conditional
- *long battery life is my top priority*

2. If someone made a camera with 20 hours of battery life, I'd buy it

- less probably conditional. Indicates preference
- *as (1), but I think it's unrealistic so I'll settle for something else*

3. If someone had made a camera with 20 hours of battery life, I'd have bought it

- Impossible past events
- *as (1), but they don't make one, so I bought something else*

More examples

2. If I wanted a camera with 20 hours of battery life, I would buy a Nikon

- *battery life is not my priority, so I'll probably buy something else*

3. If I had wanted a camera with 20 hours of battery life, I'd have bought a Nikon

- *battery life is not my priority and so I bought something other than Nikon*

Linguistic analysis of conditional types

Type 0: If + simple present --> simple present

- *If it has good battery life, you can take lots of pictures*

Type 1: If + simple present --> simple future

- *If it has good battery life, I will buy it*

Type 2: If + past --> would + infinitive

- *If it had good battery life, I would buy it*

Type 3: If + past perfect --> present perfect

- *If it had had good battery life, I would have bought it*

Simple conditional application in GATE

- Gazetteer list gives us words associated with conditionals
  - if, assuming, even if, as long as, on condition that... (positive)
  - unless (negative)
- Verb chunker segments the VPs and also gives
  - the tense of the verb
  - active or passive
  - positive or negative
- Grammar rules combine items from gazetteer with verb information to create rules for sentences

Hands-on: conditionals

- Remove all loaded applications and documents from GATE
- Load the application conditionals/resources/conditionals.gapp from the hands-on materials
- Load the document conditionals/corpus/conditional-sentences.txt, add to a corpus and run the application on it
- Check the results
- Have a look at the grammar conditional-polarity.jape and see if you can work out how the negation part works

Sample grammar for type 0 conditional

Pattern: If + simple present, simple present

```
Input: Split VG ConditionalIndicator

Rule: Conditional0
(
 {ConditionalIndicator}
 {VG.tense == SimPre}
 {VG.tense == SimPre}
 {Split}
):tag
-->
:tag.Conditional = {type = "0"}
```

{Lookup.majorType == conditional}

Verb phrases with verb in the simple present tense

Don't let the pattern span a sentence boundary

Tag the whole sentence as a conditional of type 0

Why do we do the Lookup in a separate phase?

- Why do we first find the Conditional Lookups and annotate them separately? Why not just use the Lookup annotation within the rule?
- The clue is in the Input headers
- If we use a Lookup annotation within the rule, we need to add “Lookup” to the Input headers
- What effect might this have on the rule?
- Remember that we only want to state explicitly in the rule the things we care about.
- We don't care (at this stage) which nouns occur in the sentence so we want to leave as much as possible unspecified.

Negation: adding the polarity feature

- The sentence is divided into its two verb phrases: firstPol and secondPol
- For each phase, if the value of the neg feature is “yes”, then “neg” is stored as the new value
- If the value of the neg feature is “no”, then “pos” is stored as the new value
- A new feature called “polarity” is added to the final annotation that covers the whole sentence
- The values of the two neg features (one for each VP) are added consecutively as the values of polarity, e.g. “neg” + “pos”

A more 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

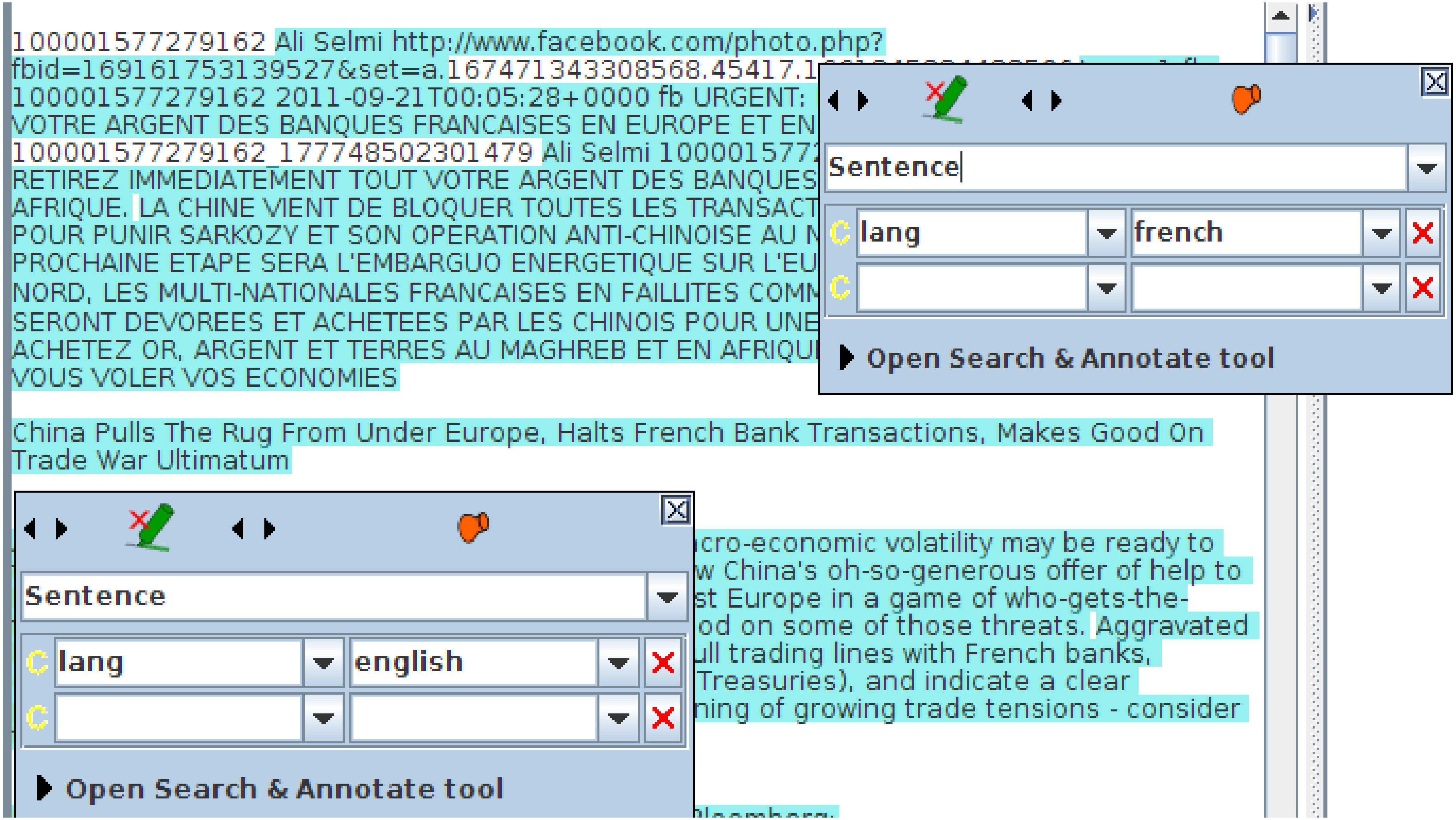
Application Stages

- Structural pre-processing, specific to social media types
- Linguistic pre-processing (including language detection), NE, term and event recognition
- Additional targeted gazetteer lookup
- JAPE grammars
- Aggregation of opinions

Linguistic pre-processing

- Language identification (per sentence) using TextCat
- Standard tokenisation, POS tagging etc using GATE
- Modified versions of ANNIE and TermRaider for NE and term recognition
- Event recognition using specially developed GATE application (e.g. band performance, economic crisis, industrial strike)

Language ID with TextCat



Basic approach for opinion finding

- 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

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) have a starting “strength” score
- These get modified by context words, e.g. adverbs, swear words, negatives and so on

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*

How do you know when 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

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**

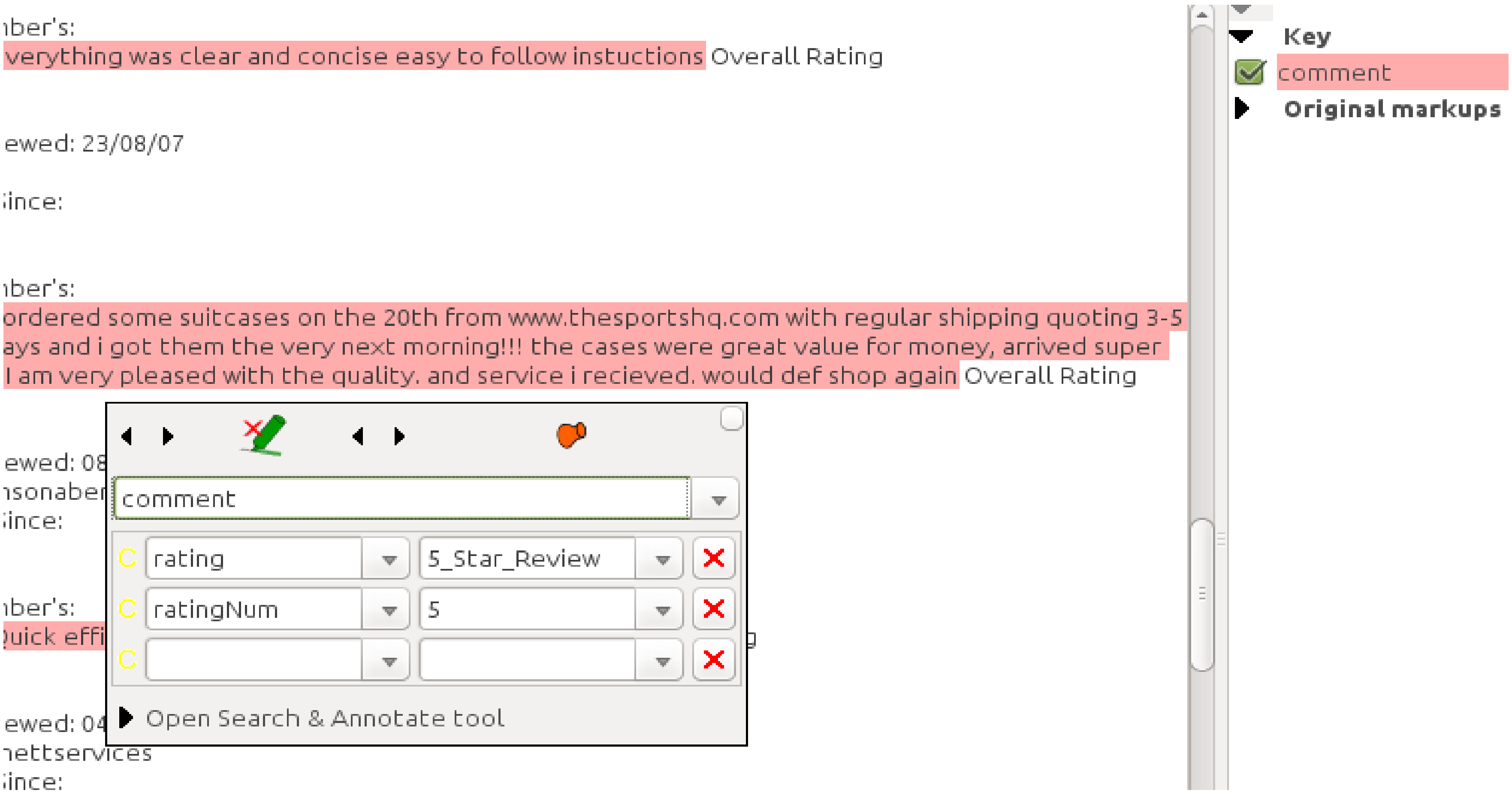
Date Reviewed: 24/01/07

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)

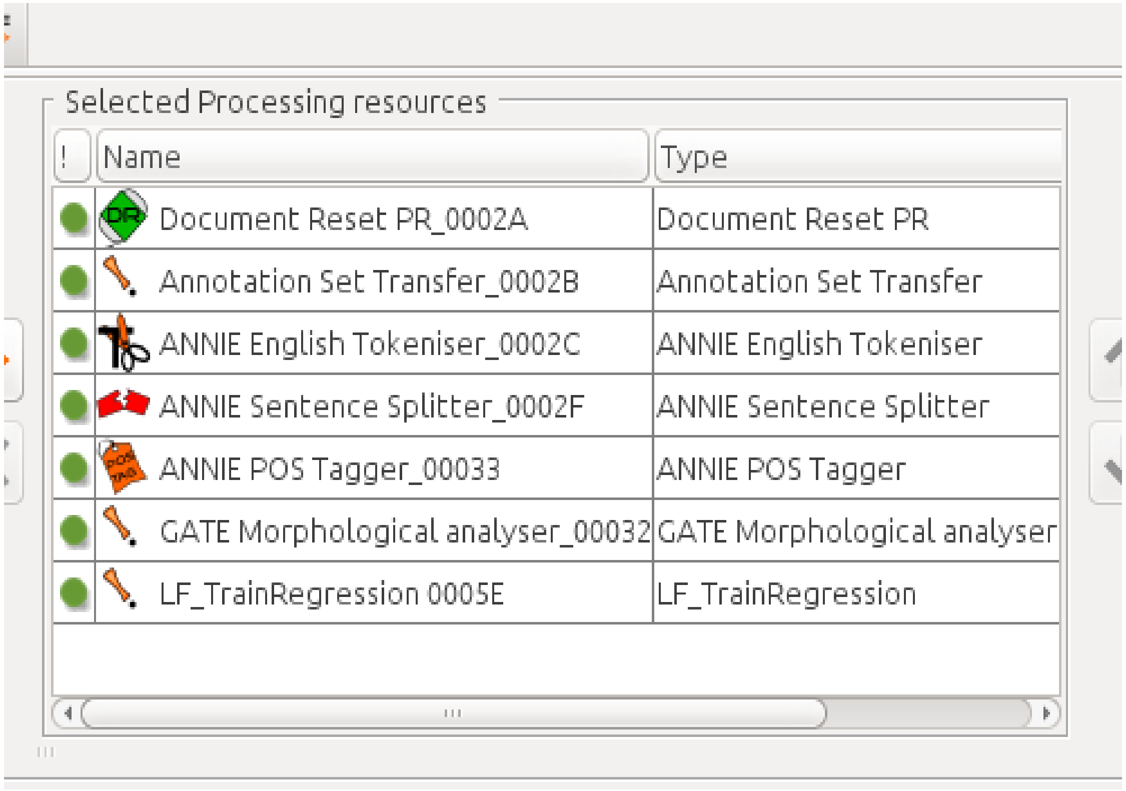
- Create the following PRs with the default init parameters:
- Document Reset PR
- Annotation Set Transfer
- ANNIE English Tokeniser
- ANNIE Sentence Splitter
- ANNIE POS Tagger
- GATE Morphological Analyser
- LF_TrainRegression
- LF_ApplyRegression
- Create a new Conditional Corpus Pipeline.

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 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 pipeline
  - English tokeniser
  - Sentence splitter
  - POS tagger
  - Morphological analyser
  - LF_TrainRegression



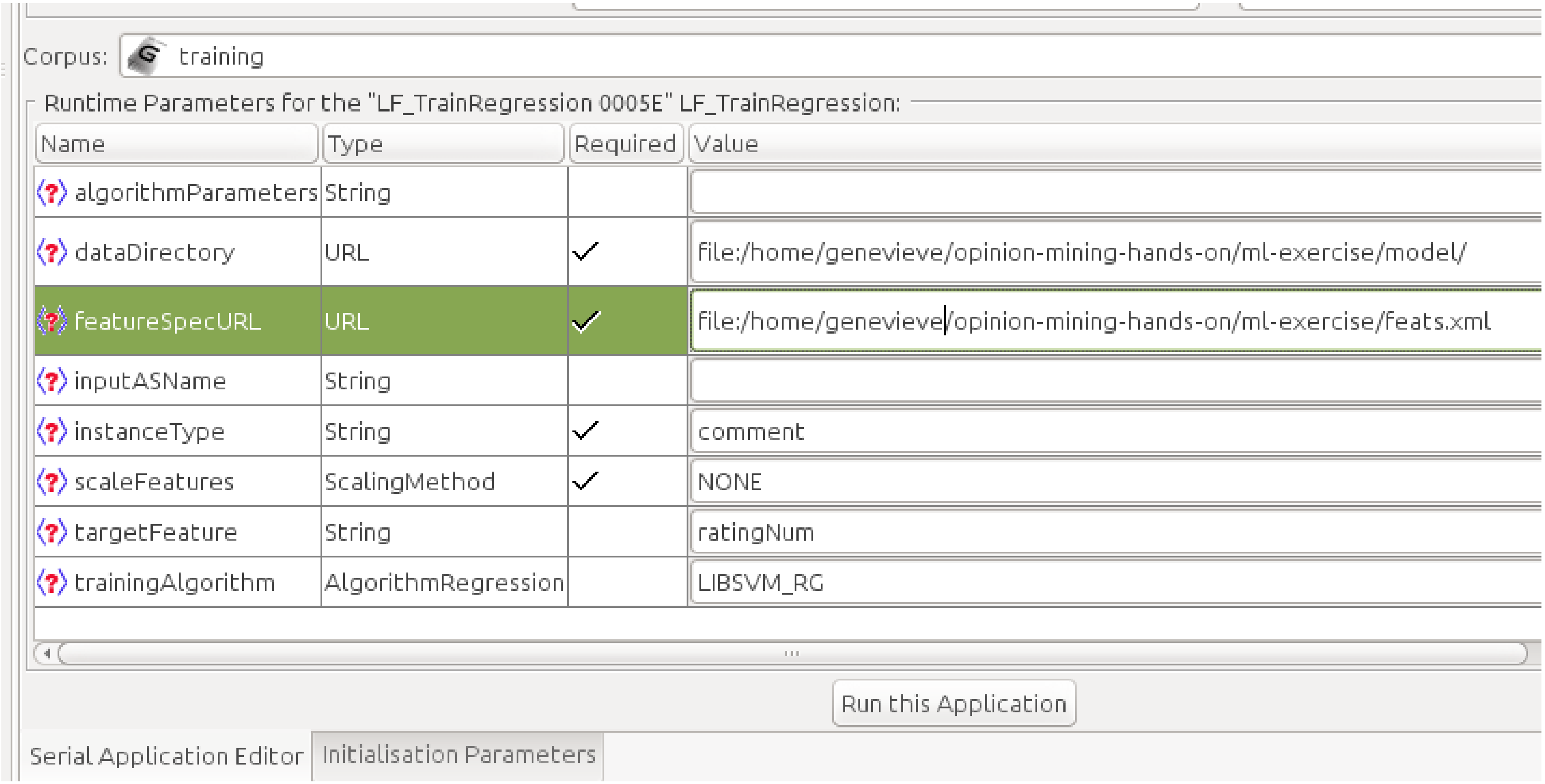
Learning Framework Parameters

- 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
- featureSpecURL is the feature specification file we inspected earlier
- 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

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

Learning Framework Parameters



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.

Applying the training model (1)

- Create a “testing” corpus and populate it from the **corpora/testing** directory.
- 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 AS 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.

LF_ApplyRegression Parameters

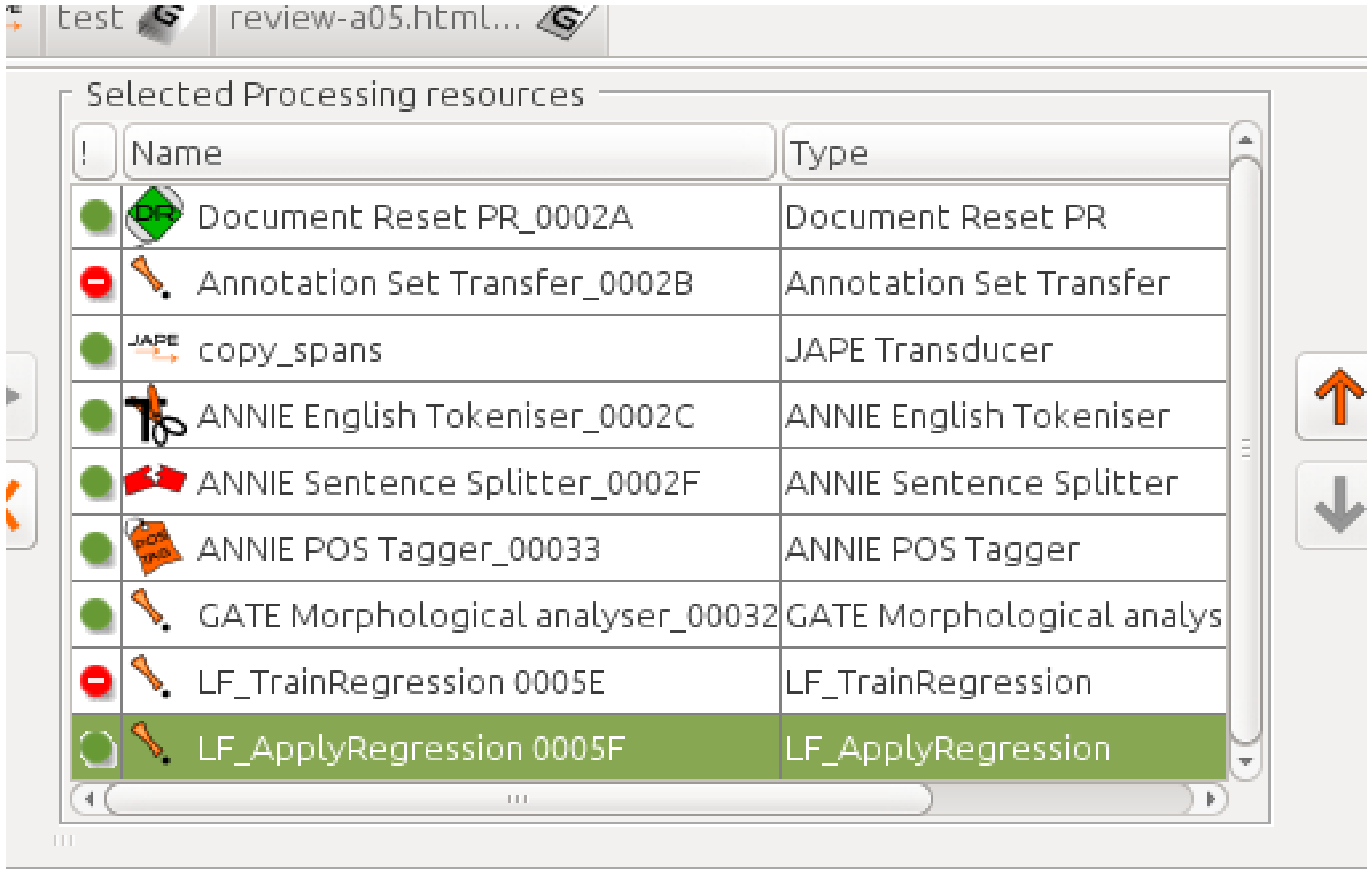
Corpus: test

Runtime Parameters for the "LF_ApplyRegression 0005F" LF_ApplyRegression:

Name	Type	Required	Value
algorithmParameters	String		
dataDirectory	URL	✓	file:/home/genevieve/opinion-mining-hands-on/ml-exercise/model/
inputASName	String		
instanceType	String	✓	comment
outputASName	String		LearningFramework
targetFeature	String		

- dataDirectory is where you saved your model during training (same location 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)

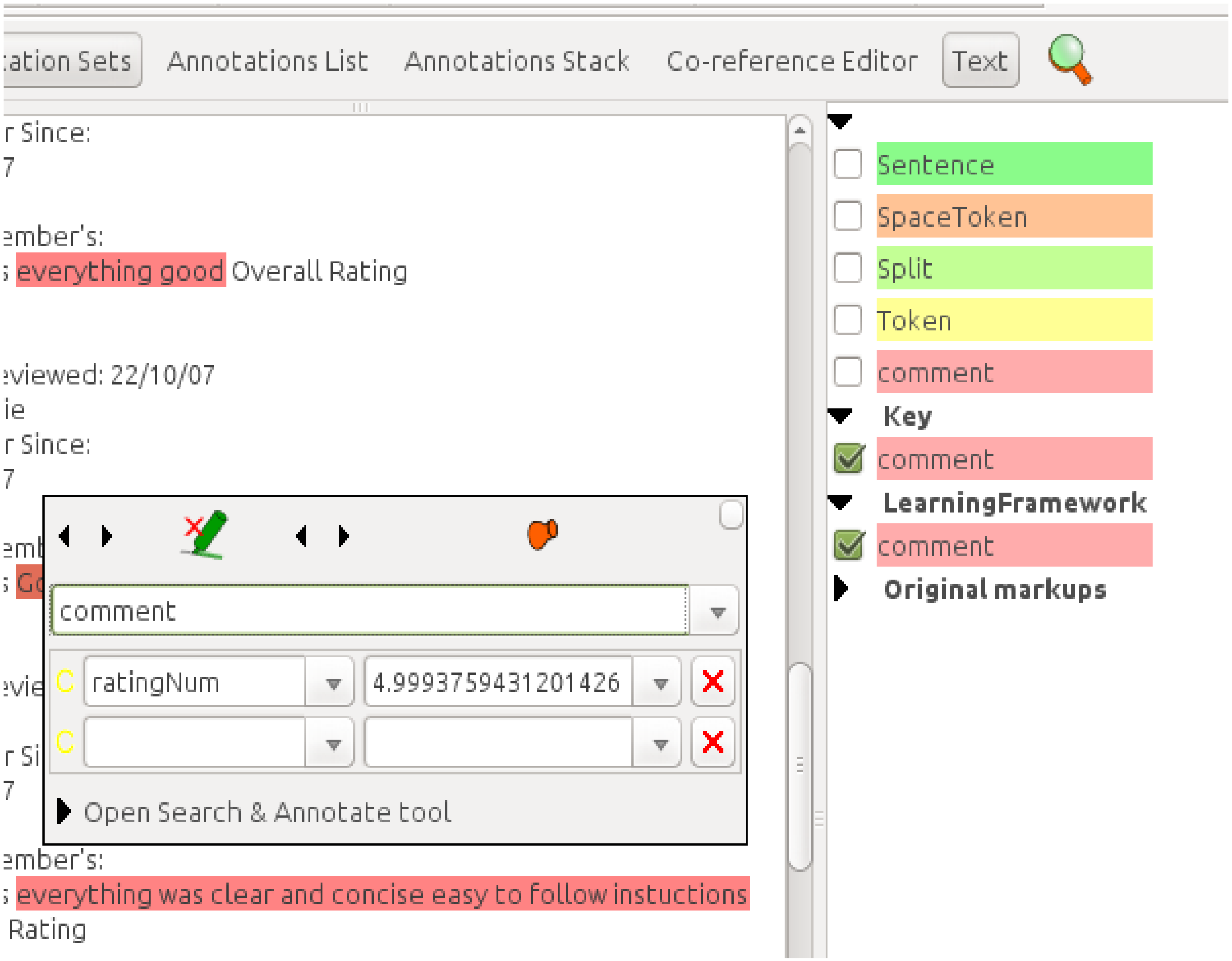


- Run the pipeline on the testing corpus

Applying the training model (3)

- Open a few documents and inspect the “comment” annotations:
- “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



Applying the training model (4)

- Note that the values are real numbers, not integers, so Corpus QA will not work
- 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.

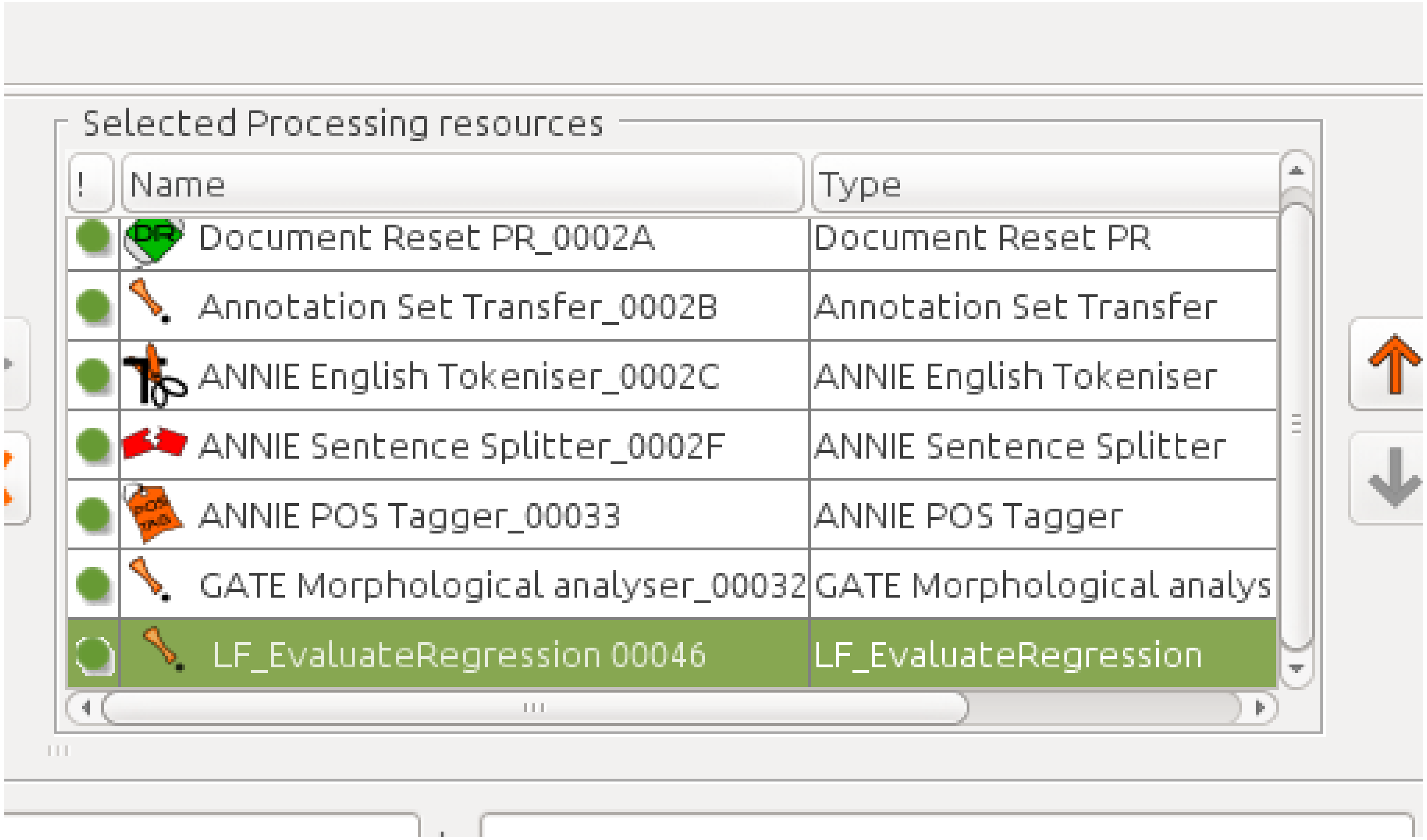
LF_EvaluateRegression

- 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).

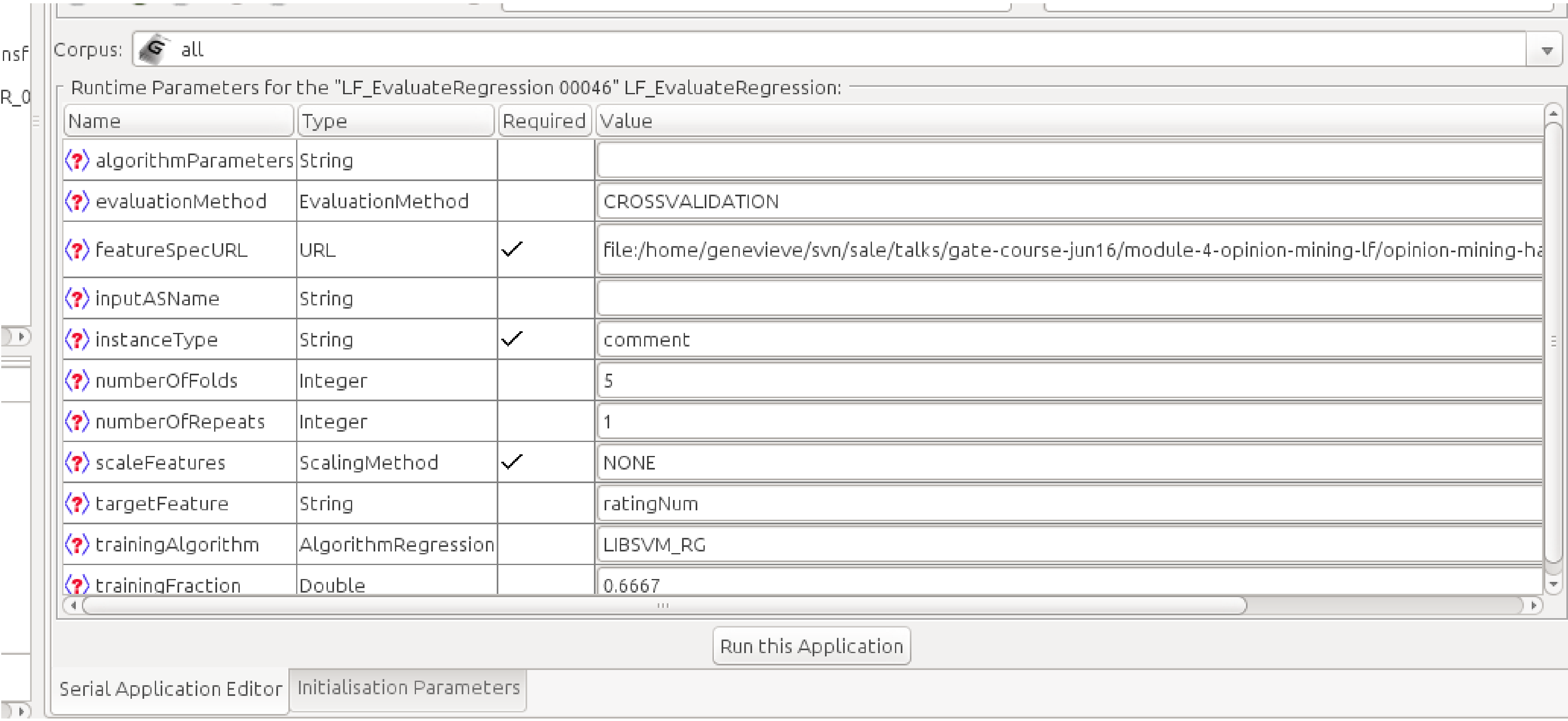
LF_EvaluateRegression

- 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).

LF_EvaluateClassification



LF_EvaluateRegression Parameters



Name	Type	Required	Value
algorithmParameters	String		
evaluationMethod	EvaluationMethod		CROSSVALIDATION
featureSpecURL	URL	✓	file:/home/genevieve/svn/sale/talks/gate-course-jun16/module-4-opinion-mining-lf/opinion-mining-ha
inputASName	String		
instanceType	String	✓	comment
numberOfFolds	Integer		5
numberOfRepeats	Integer		1
scaleFeatures	ScalingMethod	✓	NONE
targetFeature	String		ratingNum
trainingAlgorithm	AlgorithmRegression		LIBSVM_RG
trainingFraction	Double		0.6667

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?

Summary

- Introduced the concept of Opinion Mining and Sentiment Analysis
- Simple examples of rule-based and ML methods for creating OM applications
  - Examples of how deeper linguistic information can be useful
  - Practice with complex applications

More information

- There are lots of papers about this on our 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 also 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/
- More on this in Friday's applications track

More hands-on exercises

Hands-on 1: GATE sentiment tool

- Download and run the application from the GATE website

https://gate.ac.uk/applications/sentiment.html

- You need to follow the instructions carefully!
- It cannot be run without first running an application like ANNIE or TwitIE first!

Hands-on 2: Analysing tweets

- Unzip and load the bonus sentiment application **sentiment-tweets/sentiment-tweets.gapp**
- It should contain everything you need to run the application, including a small test corpus
- Run the application english-om on the document and check the results
- Try modifying the gazetteers/grammars to improve the results
- A simple improvement might be to create a Sentiment annotation for the tweet “i heart u!”
- Remember that if you modify the gazetteer which is used by an extended gazetteer, after saving it you need to reinitialise the extended gazetteer.

Hands-on 3: Using ANNIC

- Create a new Lucene datastore in GATE, using the default parameters, **but** set “AnnotationSets” parameter to exclude “Key” and “Original markups”.
- Do not double-click or “Show” the datastore. (Displaying it will slow down the following instructions.)
- Create a new empty corpus, save it to the datastore, then populate it with from the tweet-texts directory used from the social media hands-on
- Run the application english-om (from the hands-on 1) on the corpus
- When the application has finished, you can display the datastore

Hands-on 3: Using ANNIC

- 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 and want to test these

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.