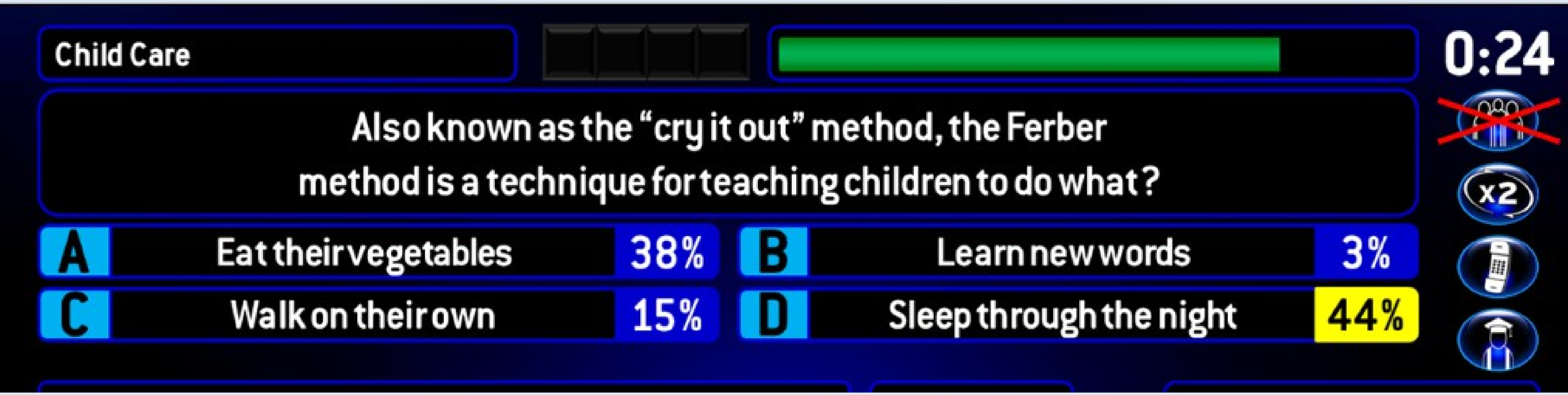
Practical Sentiment Analysis

Dr. Diana Maynard

University of Sheffield, UK

Some of the things you can learn today

Which is more accurate?



Ask the audience?

Or Phone a Friend?

How did the Royal Wedding influence the UK's health?

Can Twitter help us predict the stock market?

“Hey Jon, Derek in Scunthorpe's having a bacon and egg sandwich. Is that good for wheat futures?”

Can Twitter predict earthquakes?

Sentment Analysis Symposium, San Francisco,   October 2012

Why do deaths confuse sentiment analysis tools?

Outline

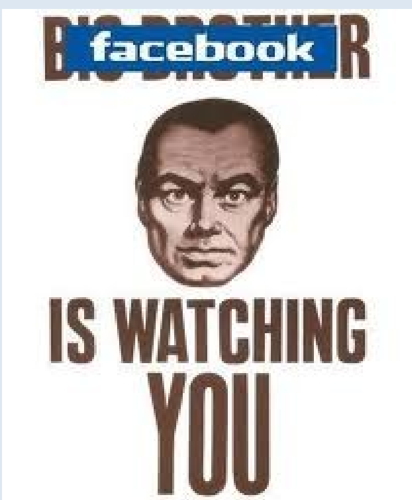
- Introduction to Opinion Mining
  - concepts and motivation, strengths and weaknesses of current systems
  - subtasks of an opinion mining system and the major challenges
- Introduction to GATE
  - main components
  - Why use GATE for opinion mining?
- Applications
  - examples of developing various real applications in GATE
  - machine learning and rule-based approaches

Part 1: Introduction to Opinion Mining

Introduction to Opinion Mining: Concepts and Motivation

The Social Web

Information, thoughts and opinions are shared prolifically these days on the social web



Drowning in information

- It can be difficult to get the relevant information out of such large volumes of data in a useful way
- Social web analysis is all about the users who are actively engaged and generate content
- Social networks are pools of a wide range of articulation methods, from simple "I like it" buttons to complete articles

Opinion Mining

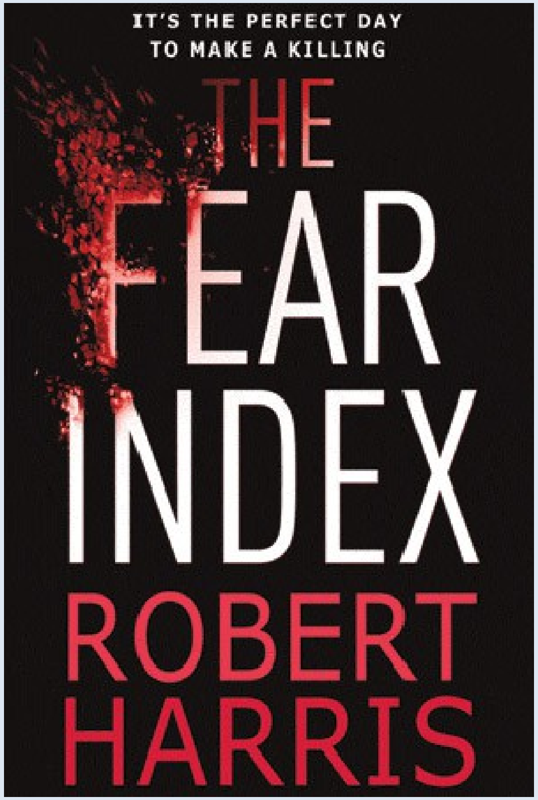
- Along with entity, topic and event recognition, opinion mining forms the cornerstone for social web analysis

Opinion mining is 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

Analysing Public Mood

- Closely related to opinion mining, is the analysis of sentiment and mood
- Mood of the Nation project at Bristol University http://geopatterns.enm.bris.ac.uk/mood/
- Mood has proved more useful than sentiment for things like stock market prediction (fluctuations are driven mainly by fear rather than by things like happiness or sadness)



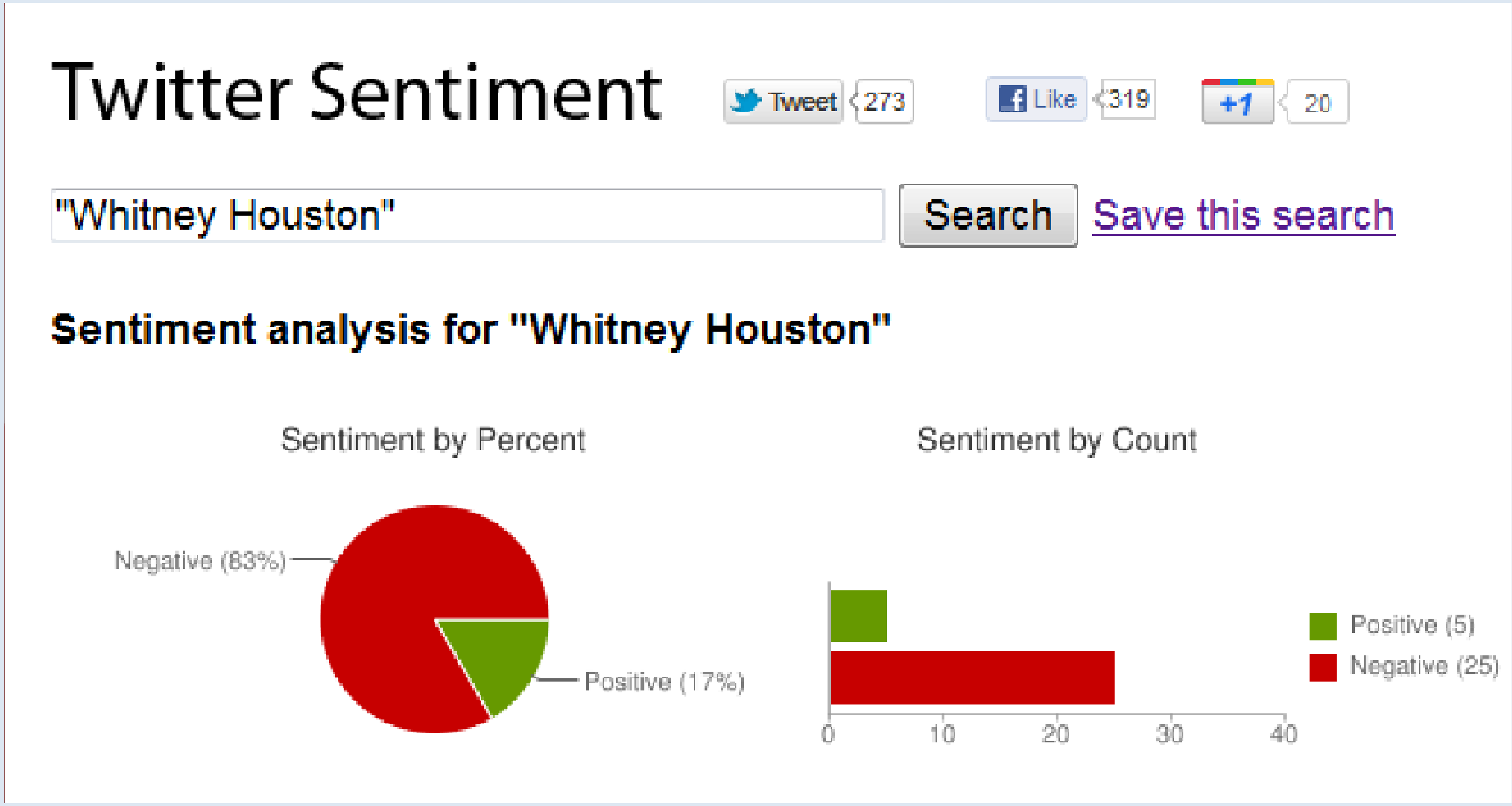
But there are lots of tools that “analyse” social media already....

- Here are some examples:
  - Sentiment140: http://www.sentiment140.com/
  - Twends: http://twendz.waggeneredstrom.com/
  - Twittratr: http://twitrratr.com/
  - SocialMention: http://socialmention.com/
  - TipTop: http://feeltiptop.com/
  - TweetFeel: http://www.tweetfeel.com/

Why not use existing online 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 very good...

Whitney Houston wasn't very popular...



Or was she?

Tweets about: "Whitney Houston"

bazzyboy25: **Whitney houston**...too soon? #CelebritiesThatLookLikeTheyStank
Posted 5 minutes ago

TeghanSimone: Radio playing **Whitney Houston**.. I swear I'm about to cry... So sad
Posted 5 minutes ago

JB3LL: hoes about to get **whitney houston**'d tonight! #TheWalkingDead
Posted 5 minutes ago

derickaadamss: “@indreamville_: Twitter I'm curious who do you think had more problems Michael Jackson or **Whitney Houston**???” <<<< **Whitney Houston**!
Posted 5 minutes ago

charlottesteer4: Listening to **Whitney Houston** loveeeee songsss <3 she's amazing <3
Posted 5 minutes ago

DionneHeraty40: @Sbarry25 The reason why **Whitney Houston** died at only 41 http://t.co/JJKRDjbj
Posted 5 minutes ago

ShortySoooFine: #musicwasbestwhen legends like James brown, Michael Jackson, **Whitney Houston** still lived.
Posted 5 minutes ago

CarlmannJohnson: Pray for Bobby Brown!!! He lost his ex-wife **Whitney Houston** and his dad Herbert Brown... Prayers up for you!!
Posted 5 minutes ago

LonelySpaceman: Is it bad that I thought **Whitney Houston** was already dead?
Posted 5 minutes ago

eatmy_CHOCLATE: My aunt in there playing **Whitney Houston** making me sad
Posted 5 minutes ago

The results for this query are: Accurate Inaccur

What did people think of the Olympics?

Twittrater's view of the Olympics

- A keyword search for Olympics shows exactly how existing systems fail to cut the mustard
- Lookup of sentiment words is not enough if
  - they're part of longer words
  - they're used in different contexts
  - the tweet itself isn't relevant
  - they're used in a negative or sarcastic sentence
  - they're ambiguous

Accuracy of twitter sentiment apps

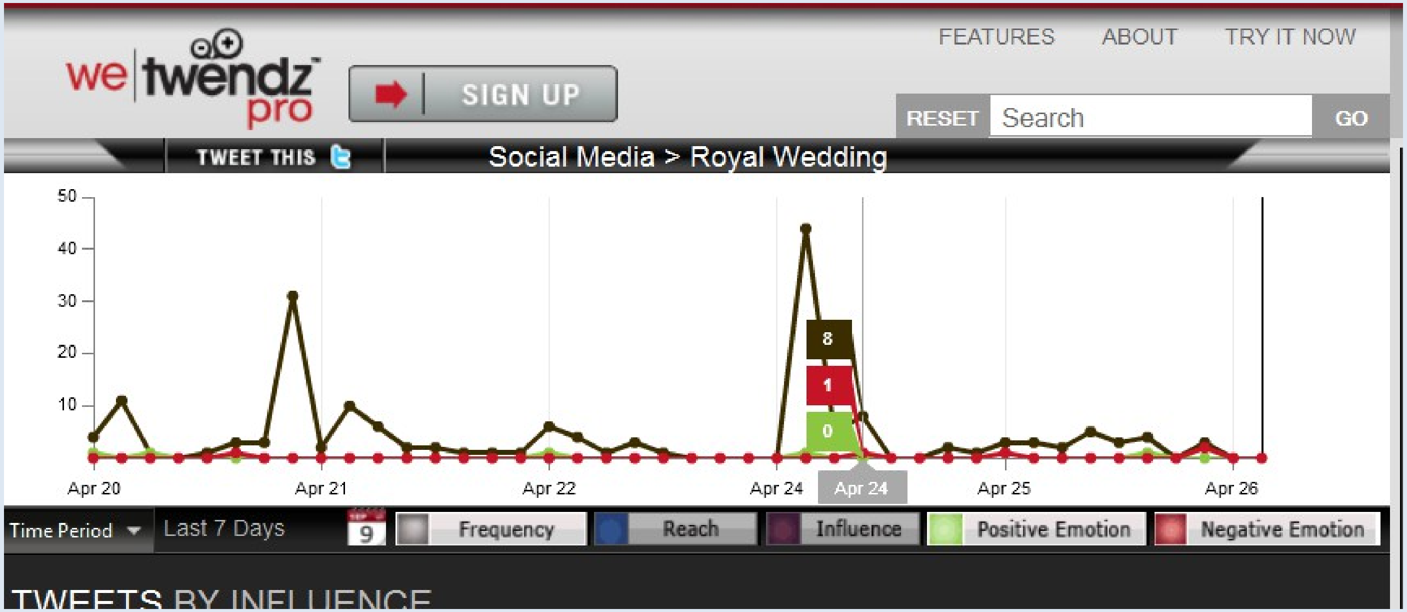
- Mine the social media sentiment apps and you'll find a huge difference of opinions
- After the Royal Wedding, a search for opinions on "Pippa Middleton" gave the following:
  - TweetFeel: 25% positive, 75% negative
  - Twendz: no results
  - TipTop: 42% positive, 11% negative
  - Twitter Sentiment: 62% positive, 38% negative

Tracking opinions over time and space

- 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.
- Twitter acitivity over 24 hours plotted on a world map http://bit.ly/SgGhIJ

Measuring impact over time

- We can measure the impact of a political entity or event on the overall sentiment about another entity or event, over the course of time.
- Aggregation of opinions over entities and events to cover sentences and documents
- Combined with time information and/or geo-locations, we can then analyse changes to opinions about the same entity/event over time, and other statistical correlations



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 have become incredibly popular in the UK solely as a result of social media.
- Pippa Middleton's bottom has its own Facebook page and twitter account, and pictures of her bottom are worth more than those of her face!

Predicting the future

Predicting other people's decisions

- It would be useful to predict what products people will buy, what films they want to see, or what political party they'll support

Predicting Presidential Candidates

- Michael Wu from Lithium did a study of sentiment data on various social web apps about presidential candidates in March 2012
- http://lithosphere.lithium.com/t5/Building-Community-the-Platform/Big-Data-Big-Prediction-Looking-through-the-Predictive-Window/ba-p/41068
- His analysis involved taking the positive sentiments minus the negative sentiments, over a 2 week period, and also including the neutral sentiments
- Neutral sentiments were weighted at 1/10 and added to the net sentiment
- He saw a close correlation between his analysis and the Gallup polls, but he warns us to be cautious...

Predictive Analysis Windows

- Predictive analytics is about trying to look into the future through the predictive window of your data.
- If you try to look outside this window, your future will look very blurry.
- It's like weather forecasting – the smaller the window, the more accurate you'll be
- The important question is not whether social media data can predict election outcome, but “how far ahead can it be predicted?”
- For something that changes very quickly like the financial market, the predictive window will be very short.
- For things that do not change as fast, the predictive window will be longer.
- For social media sentiment data, the window for election forecasting is about 1.5 to 2 weeks, (1 to be conservative).

Aggregate sentiment finding

- Aggregate sentiment finding (e.g. O'Connor et al 2010) typically uses shallow techniques based on sentiment word counting.
- Idea is that if you're only trying to find aggregates then such techniques are sufficient, even though they're far from perfect.
- Although the error rate can be high, with a fairly large number of measurements, these errors will cancel out relative to the quantity we are interested in estimating (aggregate public opinion).
- The claim is that using standard text analytics techniques on such data can actually be harmful, because they're designed to optimise per-document classification accuracy rather than assessing aggregate population proportions.
- Their method shows some correlation with public sentiment polls but they conclude that better opinion mining would be beneficial.

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

Introduction to Opinion Mining: Subtasks and Challenges

Opinion Mining Subtasks

- **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?
- Parsing is the obvious solution to this, but it doesn't work very well with degraded texts.
- More shallow forms of analysis may be necessary.


WHO WANTS TO BE A
MILLIONAIRE
WHO WANTS TO BE A

Fenway Park is the home ground of which Major League baseball team?

A: Boston Red Sox

B: New York Mets

C: San Francisco Giants

D: Houston Astros

What term is used in cricket when a bowler oversteps the line before releasing the ball?

A: no delivery

B: illegal ball

C: no ball

D: illegal pitch

How long does a goalball match last?

A: 2 x 12 minutes

B: 4 x 10 minutes

C: 2 x 20 minutes

D: 4 x 15 minutes

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
- Asking the first question to a US audience should get a majority correct answer
- Asking the second question to a US audience might not, though it certainly would in the UK
- Asking the third question to anyone except a goalball player would probably not get a majority correct answer

So 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

- **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, containing clusters of interests and likes/dislikes
- **Reputation (global) 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.
- But be wary of extending these to opinions about different topics.
  – Your friend who likes the same kind of books as you might not like the same kind of cameras as you.

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)

Reputation Profiling: Replab

- Evaluation campaign for Online Reputation Management Systems
- Details at http://www.limosine-project.eu/events/replab2012
- Aims to answer questions like:
  - What is the general state of opinion about a company/individual in online media?
  - What are its perceived strengths and weaknesses as compared to its peers/competitors?
  - How is the company positioned with respect to its strategic market?
  - Can incoming threats to its reputation be detected early enough to be neutralized before they effectively affect reputation?

Reputation is not the same as opinion

- Polarity here refers to “brand management” rather than standard sentiment
- For example, a negative opinion about one company might mean positive polarity for a rival company
- Similarly, a statement like “Shares in Facebook dropped by 2% today” displays neutral opinion, but negative reputaton polarity for Facebook.

Systems and results

- The GATE system was based on ML
  - For the polarity detection, it did not use sentiment lexicons but standard NE tools plus an emoticon processor as features
  - Words were excluded as features as they led to a 5% performance drop, possibly due to small training set
- The top system, Daedalus, used their existing multilingual SA software, Stylus.
  - Used sentiment lookup and rules, morphosyntactic analysis, fine-grained negation detection, aggregation, etc.

Sentiment Lexicons

- There are lots of sentiment lexicons out there, e.g. SentiWordNet, Bing Liu lexicon, MPQA, LIWC
- More info at http://sentiment.christopherpotts.net/lexicons.html
- But sentiment words are context-dependent and ambiguous
  - "a long dress" vs "a long walk" vs "a long battery lfe"
  - "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

Try some different lexicons

- http://sentiment.christopherpotts.net/lexicon/ Get sentiment scores for single words from a variety of sentiment lexicons
- http://sentiment.christopherpotts.net/textscores/ Show how a variety of lexicons score novel texts

Find the hidden deer…

One of the trickiest tasks in opinion mining is spotting the hidden meaning in a piece of text.

Irony and sarcasm

- *The now abandoned HP TouchPad is officially the hottest piece of consumer electronics on Amazon.*
- *Life's too short, so be sure to read as many articles about celebrity breakups as possible.*
- *I had never seen snow in Holland before but thanks to twitter and facebook I now know what it looks like. Thanks guys, awesome!*
- *On a bright note if downing gets injured we have Henderson to come in.*
- *Am glad 10 day forecast calling for lots of rain/cool temps. Was getting tired sun & dry conditions*

How do you know when someone is being sarcastic?

- Use of hashtags in tweets such as #sarcasm
- 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

lol letting a baby goat walk on me probably wasn't the best idea. Those hooves felt great. #sarcasm

There's no better start into the working week than a construction site right beneath your office. Sounds a bit like Neubauten.

How else can you deal with it?

- 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

Ambiguity in tweets

- Social media posts can be ambiguous, for a number of reasons
- Ambiguity between conversation participants:

"I love Eminem" "I like Skittles better." "No, the rapper you idiot.." "You're the idiot! What's good about a M&M wrapper?!"

- Ambiguity requiring current local context

"There is a lot of dirt on Jimmy Savile."

Entity ambiguity

I like how "RIP Fergie" is trending because of football and half the population of Twitter think that one of the Black Eyed Peas has died.

- We can sometimes disambiguate entities based on context (see Part 3 of this tutorial)
- But it's hard to resolve (even for a person) when there's no contextual reference.

Part 2: Opinion Mining and GATE

GATE (General Architecture for Text Engineering)

- Our examples mostly use GATE – tool for Language Engineering in development in Sheffield since 2000.

- GATE includes:
  - **components** for language processing, e.g. parsers, machine learning tools, stemmers, IR tools, IE components for various languages...
  - tools for **visualising** and **manipulating** text, annotations, ontologies, parse trees, etc.
  - **various information extraction** tools
  - **evaluation** and **benchmarking** tools
- More info and freely available at http://gate.ac.uk

GATE: the Swiss Army Knife of NLP

- An attachment for almost every eventuality
- Some are hard to prise open
- Some are useful, but you might have to put up with a bit of clunkiness
- Some will only be useful once in a lifetime, but you're glad to have them just in case.
- There are many imitations, but nothing like the real thing :-)

Architectural principles

- Non-prescriptive, theory neutral (strength and weakness)
- Re-use, interoperation, not reimplementation (e.g. diverse XML support, integration of Protégé, Jena, Yale...)
- (Almost) everything is a component, and component sets are user-extendable
- (Almost) all operations are available both from API and GUI

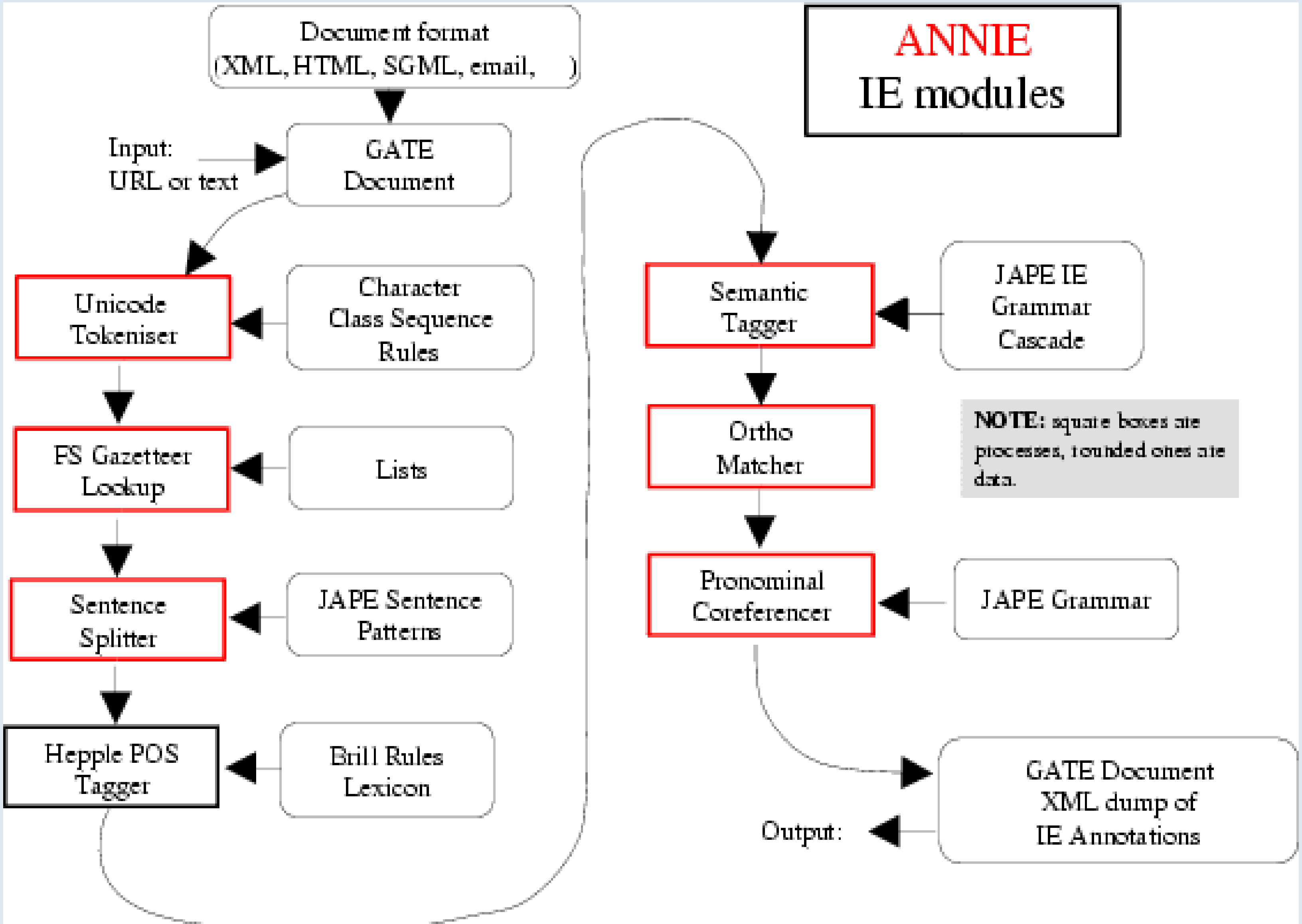
GATE components

- **Language Resources** (LRs), e.g. lexicons, corpora, ontologies
- **Processing Resources** (PRs), e.g. parsers, generators, taggers
- **Visual Resources** (VRs), i.e. visualisation and editing components
- Algorithms are separated from the data, which means:
  - the two can be developed independently by users with different expertise.
  - alternative resources of one type can be used without affecting the other, e.g. a different visual resource can be used with the same language resource

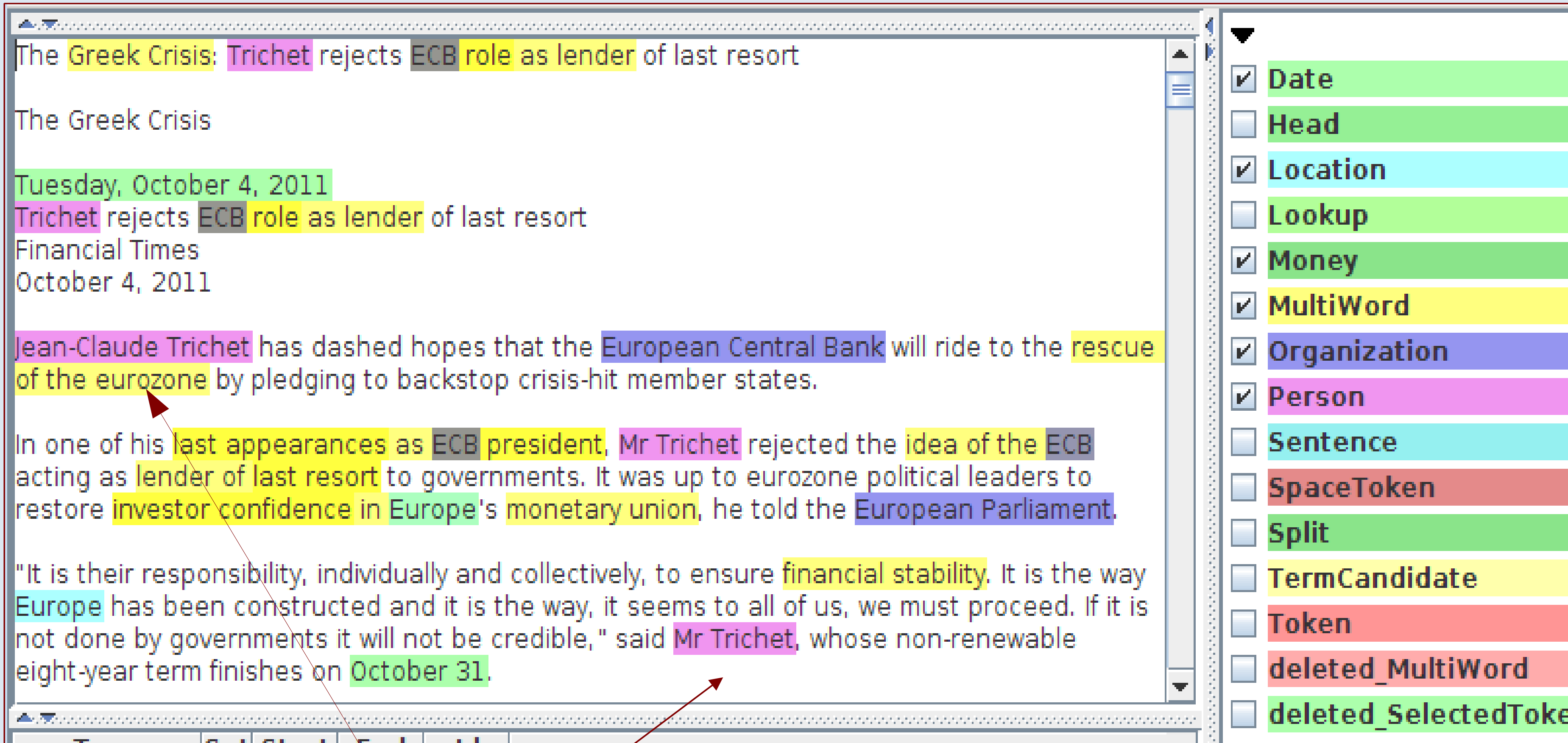
ANNIE

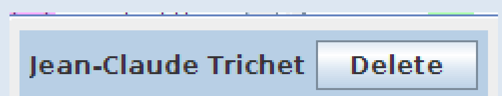
- **ANNIE** is GATE's rule-based IE system
- It uses the language engineering approach (though we also have tools in GATE for ML)
- Distributed as part of GATE
- Uses a finite-state pattern-action rule language, JAPE
- ANNIE contains a reusable and easily extendable set of components:
  - generic preprocessing components for tokenisation, sentence splitting etc
  - components for performing NE on general open domain text

ANNIE Modules



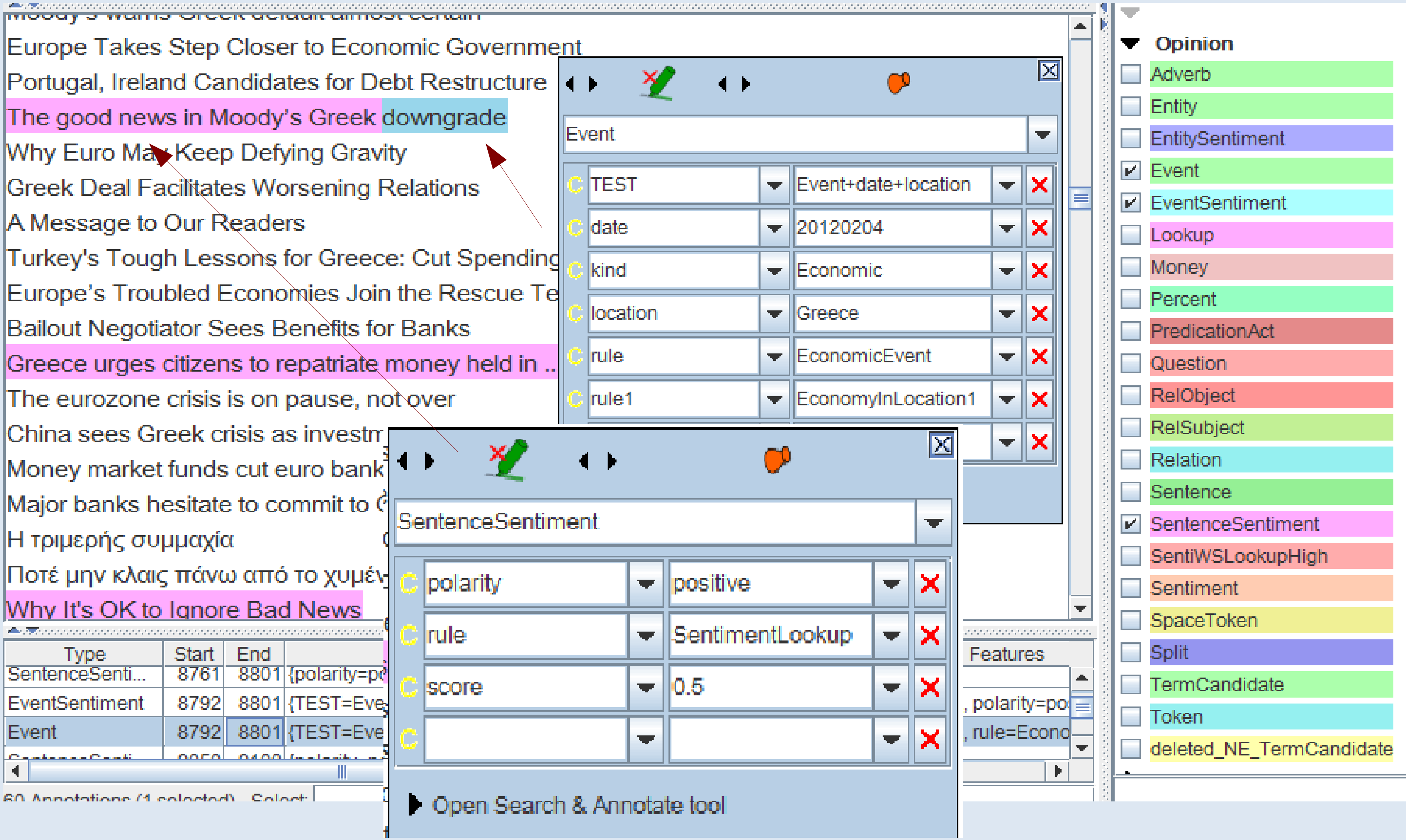
Document annotated by ANNIE





items in the co-reference chain

Opinions on Greek Crisis



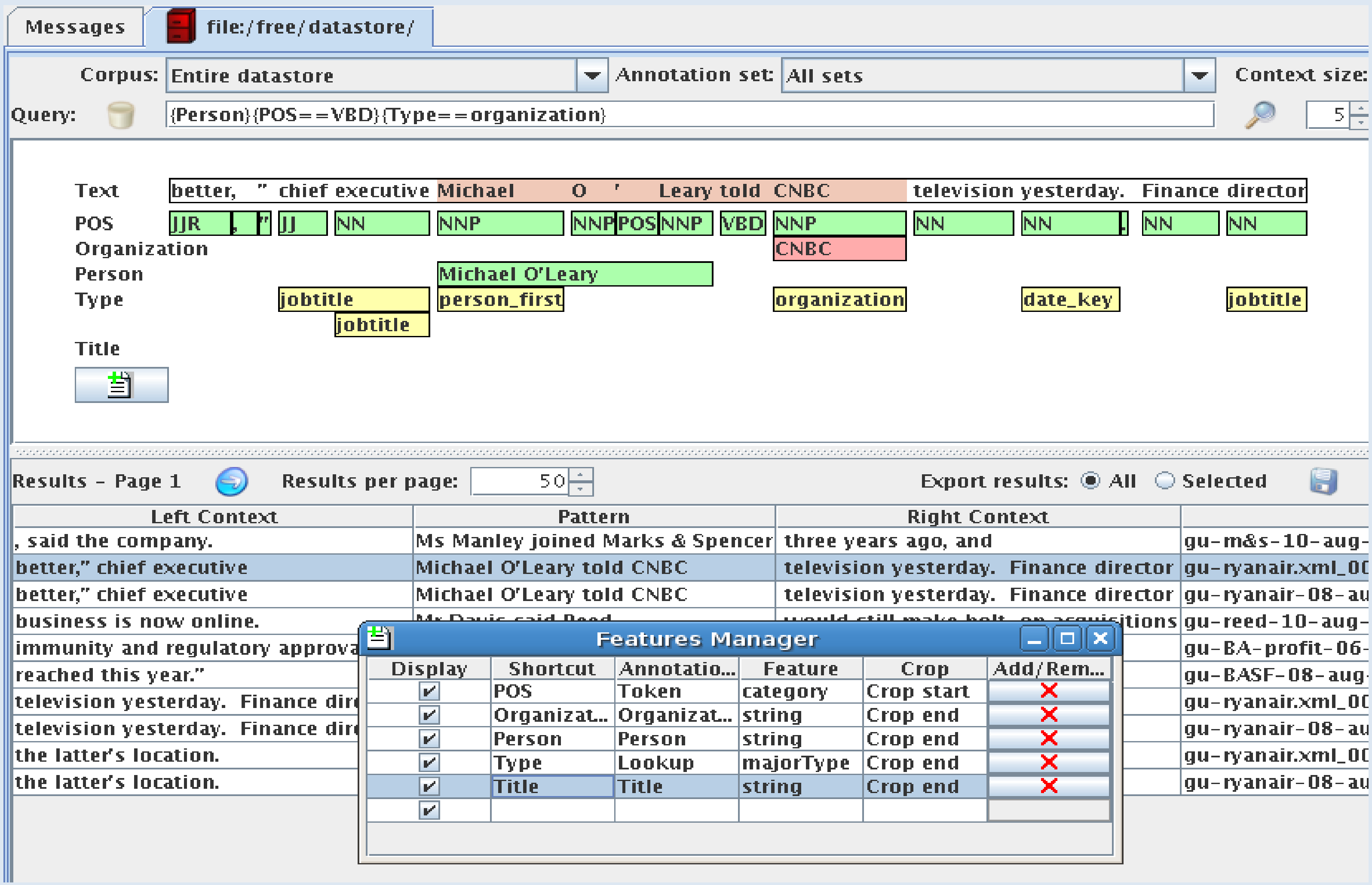
GATE for processing social media

- GATE is a great tool for opinion mining on social media
- Document format analysis separates content from metadata
- Linguistic pre-processing (including specialised Twitter components)
- NE recognition which can be easily tailored to a domain
- Support for rule-based and/or ML components for opinion finding
- Mix and match of different tools in a single pipeline
- Tools for collaborative manual annotation and automatic evaluation
- Tools for corpus analysis help identify and fix errors

Corpus analysis tools

- Corpus analysis tools enable you to look at the results of processing and make sense of them manually
- In GATE, we have a tool called ANNIC which lets you analyse annotations in context.
- Like a KWIC index but works over annotations as well as just strings
- Enables you to search and analyse a whole corpus without knowing a priori what appears specifically in which document
- This is especially useful in a corpus of tweets where each document represents a single tweet
- Demo: http://gate.ac.uk/demos/annic2008/Annic-only.htm

ANNIC example



Pattern examples

- {Party}
- {Affect}
- {Lookup.majorType == negation} ({Token})*4 {Lookup.majorType == "vote"}{Lookup.majorType == "party"}
- {Token.string == "I"} ({Token})*4 {Lookup.majorType == "vote"} {Lookup.majorType == "party"}
- {Person} ({Token})*4 {Lookup.majorType == "vote"}{Lookup.majorType == "party"}
- {Affect}   ({Token})*5   {Lookup.majorType == "candidate"}
- {Vote} ({Token})*5   {Lookup.majorType == "candidate"}

3. Applications

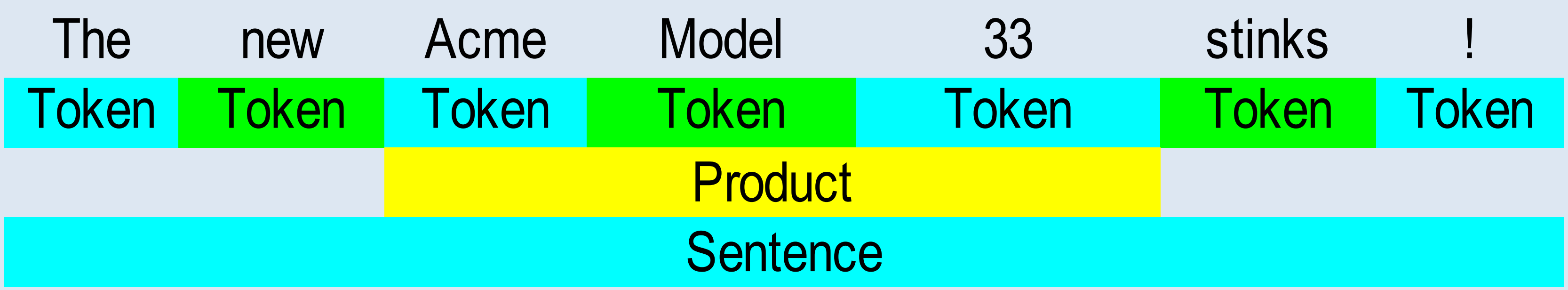
Methods for opinion mining: Machine learning

What is Machine learning?

- Automating the process of inferring new data from existing data
- In GATE, that means creating annotations or adding features to annotations by learning how they relate to other annotations

Learning a pattern

- For example, we have Token annotations with string features and Product annotations



- ML could learn that a Product close to the Token “stinks” expresses a negative sentiment, then add a polarity=“negative” feature to the Sentence.

How is that better than a rule-based approach?

- Not necessarily better, just different
- People are better at writing rules for some things, ML algorithms are better at finding some things
- With ML you don't have to create all the rules, **but** you have to manually annotate a training corpus—or get someone else to do it!
- Rule-based approaches (such as JAPE) and ML work well together; in GATE, JAPE is often used extensively to prepare data for ML.

Terminology: Instances

- Instances are cases that may be learned
- Every instance is a decision for the ML algorithm to make
- To which class does this instance belong?
  - "California" → Location
  - "This product stinks" → polarity=negative

Terminology: Attributes

- Attributes are pieces of information that we already know about instances (sometimes called “features” in machine learning literature).
- These can be GATE annotations, or annotation features that will be known before the ML algorithm is applied to new data
- Examples
  - Token.string == “stinks”
  - Token.kind == “punctuation”
  - Sentence contains Product

Terminology: Classes

- The class is what we want to learn
- Suppose we want to find opinions: for every Sentence instance, the question is “What kind of opinion does this express?” and the classes are *positive*, *negative*, *neutral*, and *none*.

ML Tasks

- GATE supports 3 types of ML tasks:
  - chunk recognition (named entity recognition, NP chunking)
  - text classification (sentiment classification, POS tagging)
  - relation annotation
- Most opinion mining tasks fall under text classification

Training

- Training involves presenting data to the ML algorithm from which it creates a model
- The training data consist of instances that have been annotated with correct classes as well as attributes
- Models are representations of decision-making processes that allow the ML algorithm to classify each instance based on its attributes

Application

- When the ML algorithm is applied, it creates new class annotations on data using the model
- The corpus it is applied to must contain the required attribute annotations
- The machine learner will work best if the application data is similar to the training data

Evaluation

- We want to know how good our machine learner is before we use it for a real task
- Therefore we apply it to some data for which we already have class annotations
  - the “right answers”, sometimes called “gold standard”
- If the machine learner creates the same annotations as the gold standard, then we know it is performing well
- GATE's ML PR has a built-in evaluation mode that splits the corpus into training and test sets and cross-validates them

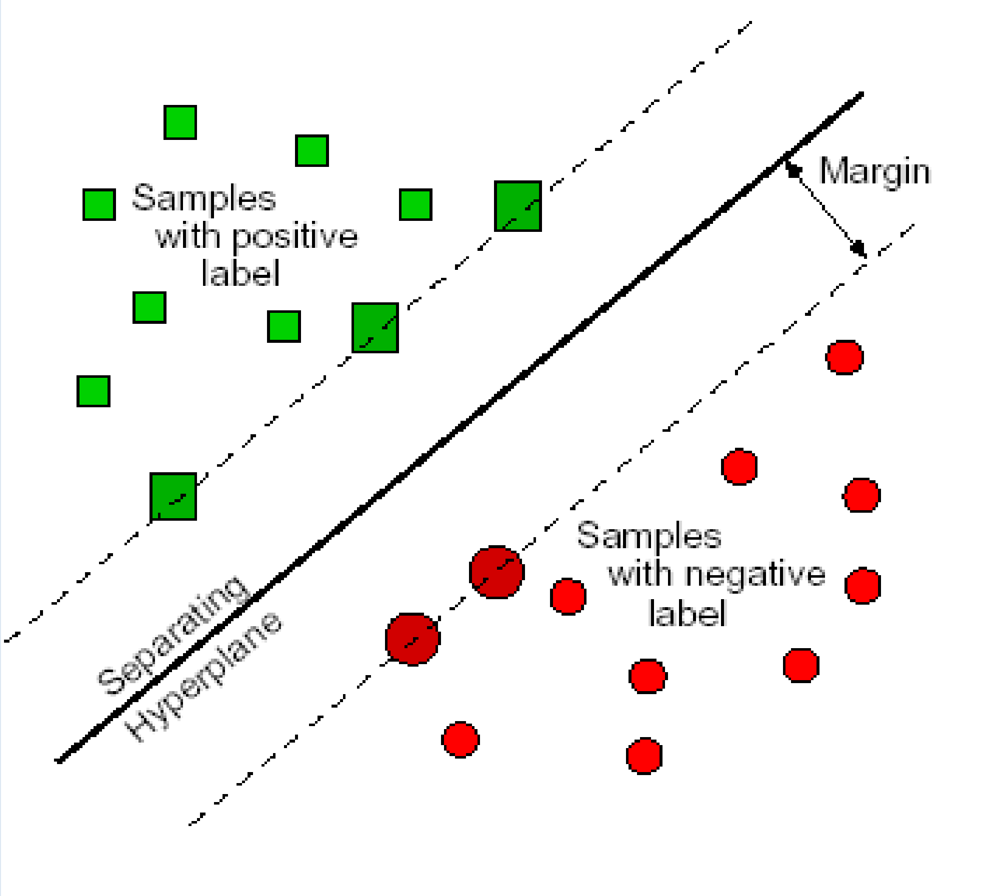
Perceptron and PAUM

- Perceptron is one of the oldest ML methods (invented in the 50s!)
- Like SVM (which will be covered later), it determines a hyperplane separator between the data points
- Theoretically SVM works a little better because it calculates the optimal separator, but in practice, however, there is usually little difference, and Perceptron is a lot faster!

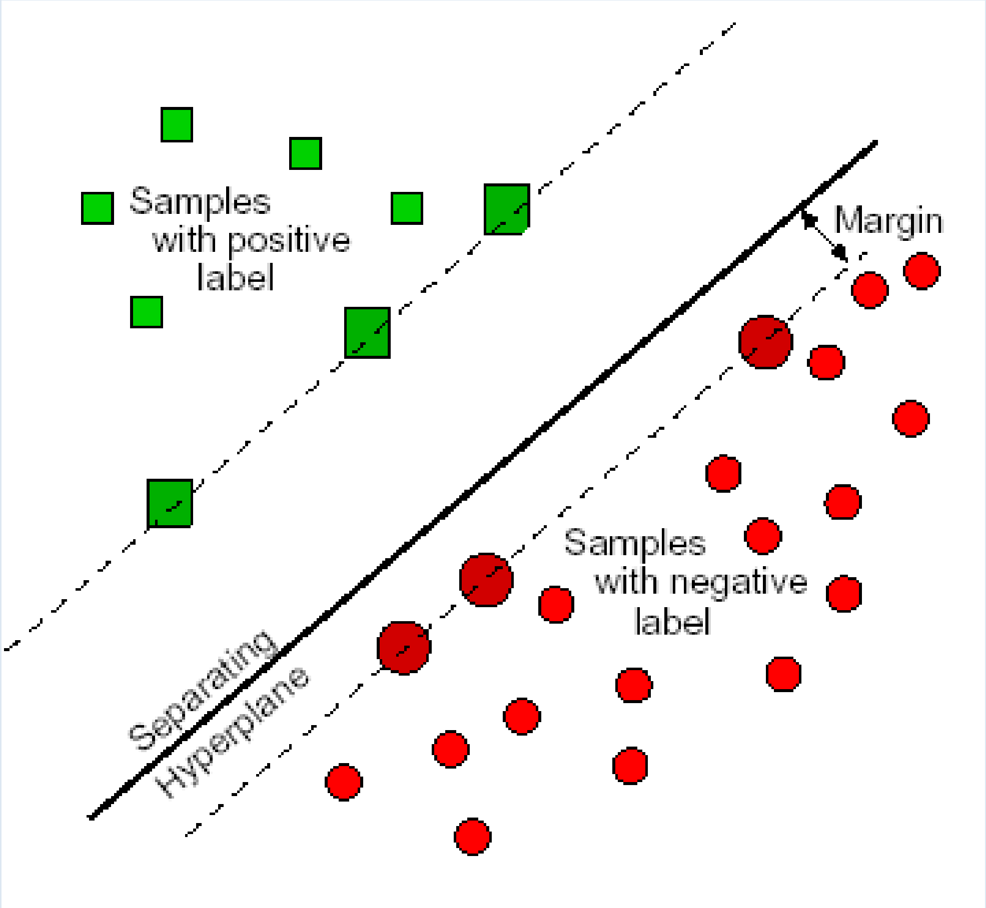
Perceptron Algorithm with Uneven Margins (PAUM)

- Both Perceptron and SVM implement “uneven margins”
- This means that it doesn't position the separator centred between the points, but more towards one side

Even Margins

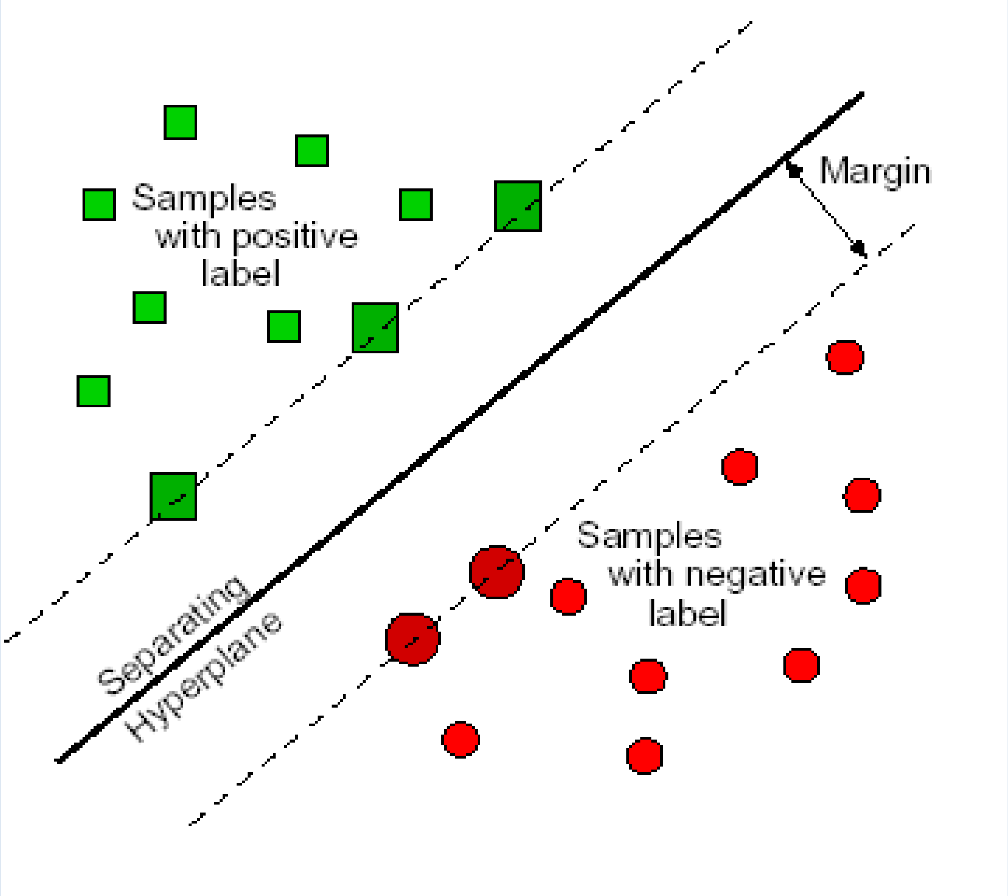


Uneven Margins



Support Vector Machines

- Like Perceptron, try to find a hyperplane that separates data
- But the goal here is to **maximize** the separation between the two classes
- Wider margin = greater generalisation

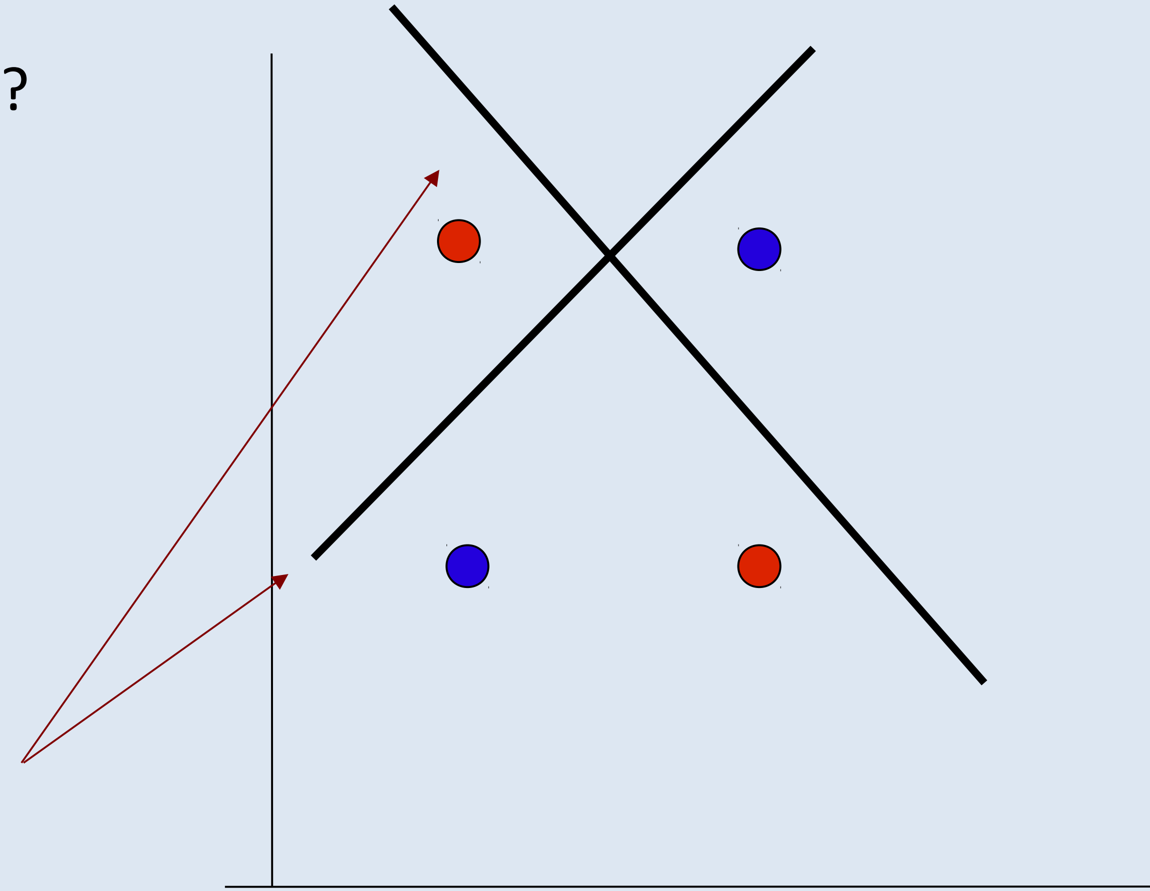


Support Vector Machines

- The points near the decision boundary are the “support vectors” (removing them would change boundary)
- The farther points are not important for decision-making
- What if you can't split the data neatly?
  - Soft boundary methods exist for imperfect solutions
  - However linear separator may be completely unsuitable

Support Vector Machines

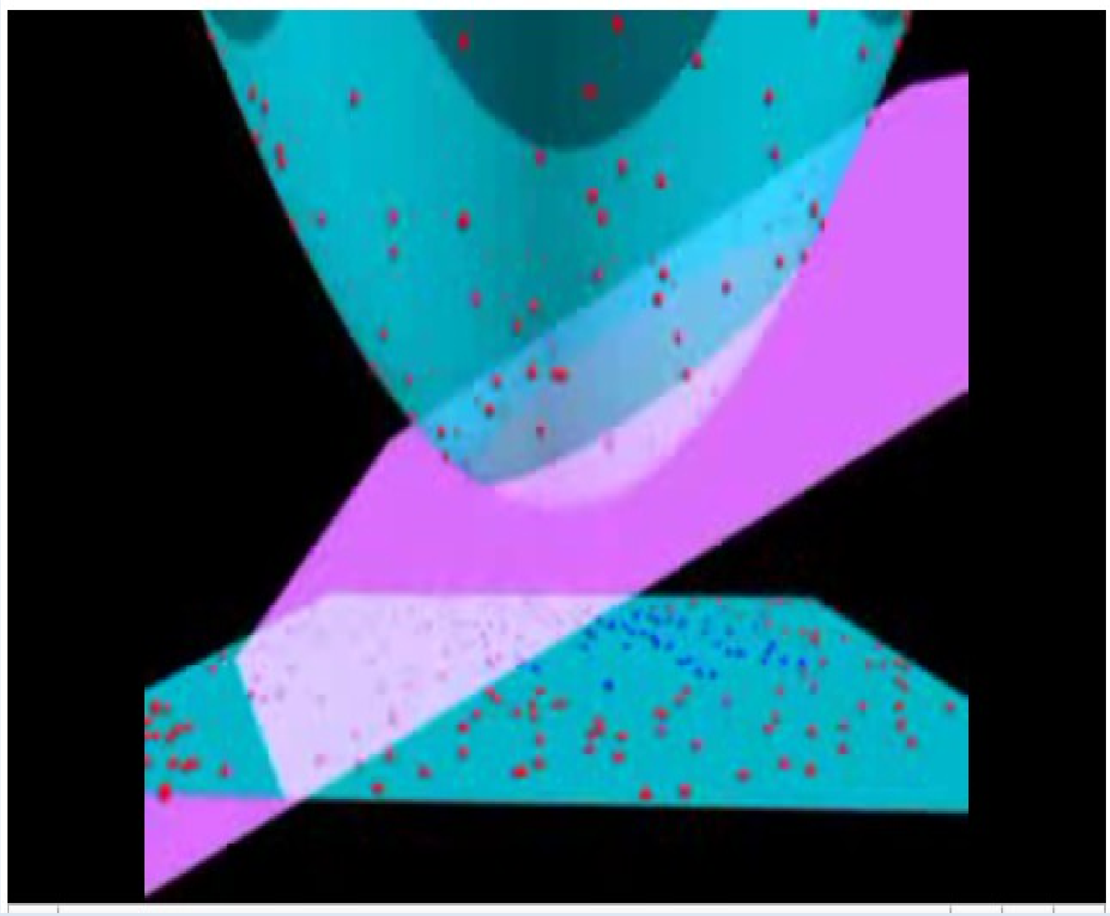
- What if there is no separating hyperplane?



They do not work!

Kernel Trick

- Map data into different dimensionality
- http://www.youtube.com/watch?v=3liCbRZPrZA
- As shown in the video, due to polynomial kernel elliptical separators can be created nevertheless.
- Now the points are separable!



Kernel Trick in GATE and NLP

- Binomial kernel allows curved and elliptical separators to be created
- These are commonly used in language processing and are found to be successful
- In GATE, linear and polynomial kernels are implemented in Batch Learning PR's SVM engine

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

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

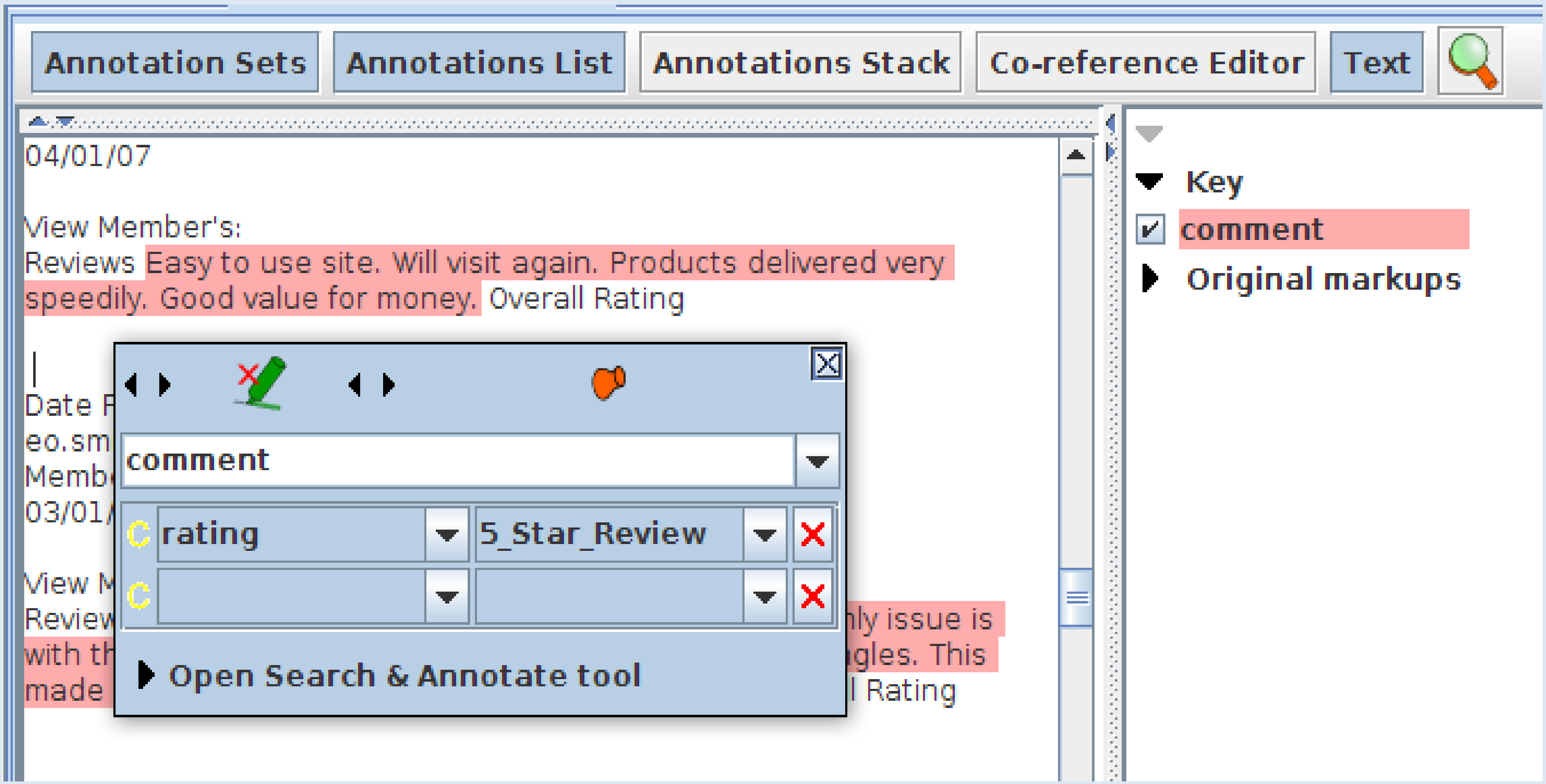
Dbeach135 **Member Since:** 24/01/07 **View Member's:** Reviews	Jessops not only failed to complete the next day delivery, the item sent, a digital picture frame did not meet their specification. We ordered it as they claimed on their website that it accepted XD cards. This however was not the case. 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**

Case study 1:
Opinion Mining in Consumer Reviews

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
- We develop an application that runs a set of NLP components to provide ML instance attributes, and train the classifier

Annotated review



ML configuration

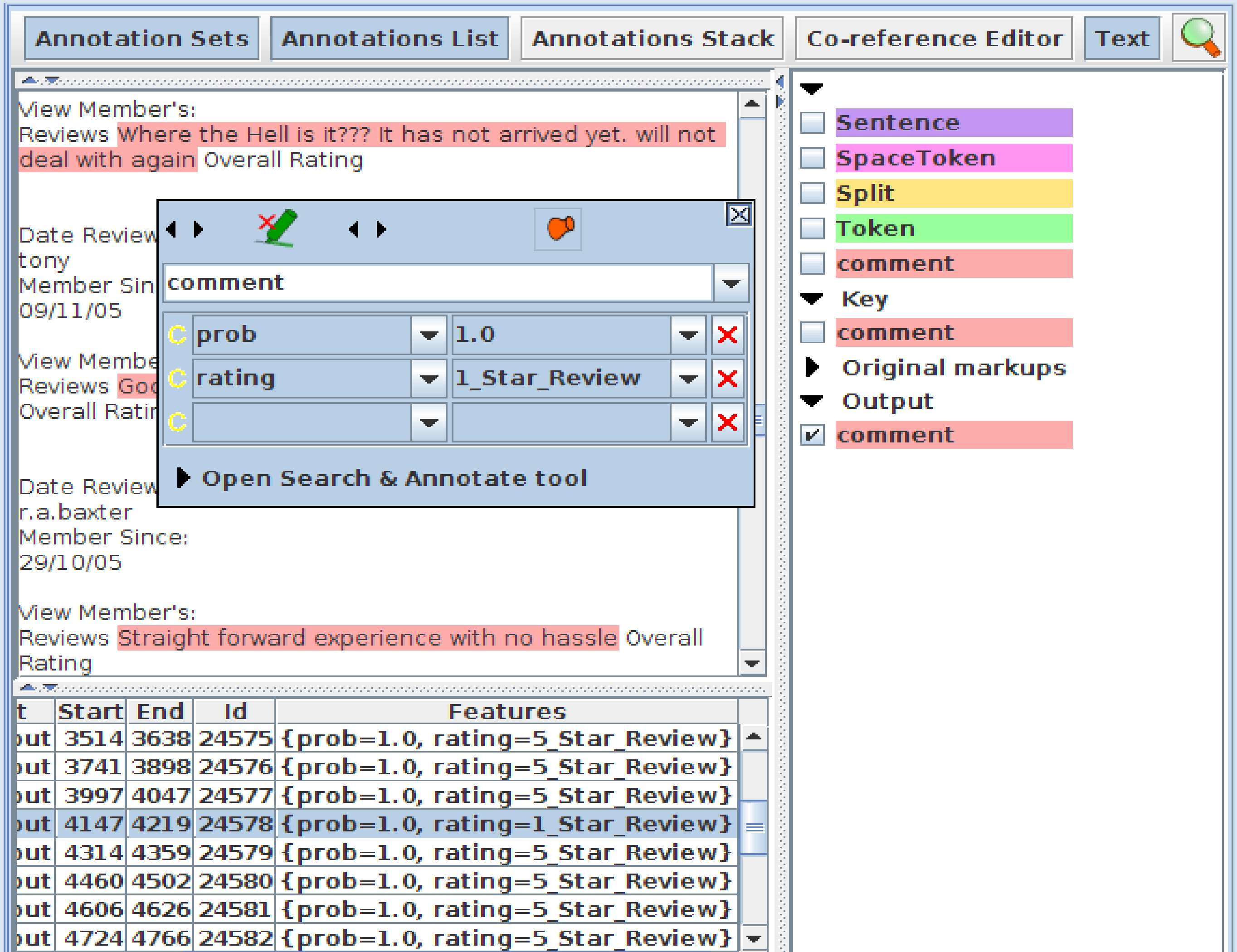
- For this application, we used SVM (we would probably use PAUM now)
- Attributes: bag of lemmatised words (unigrams of lemmata) inside each comment annotation

Applying the training model

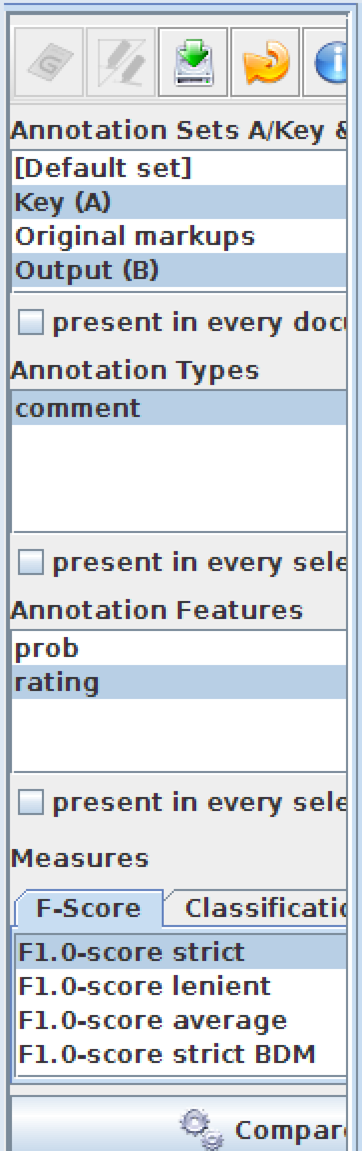
- To apply the classifier to our test corpus, we need to have comment annotations *without* rating features on the default AS
- These will give us the instances to classify
- A simple JAPE Transducer can do this
- When the pipeline is run, the classifier will get instances (*comment* annotations) and attributes from the default AS and put instances with classes (*rating* features) in the Output AS

  – Key set = user ratings
  – default set = instances with no classes
  – Output set = instances with ML classes

Annotation Results



Evaluation: Corpus QA tool in GATE



Results

Corpus statistics | Document statistics

Annotation	Match	Only A	Only B	Overlap	Rec.B/A	Prec.B/A	F1.0-s.
comment	79	20	20	0	0.80	0.80	0.80
Macro summary					0.80	0.80	0.80
Micro summary	79	20	20	0	0.80	0.80	0.80

Cohen's Kappa and confusion matrices

- We can also use the Cohen's Kappa measure to show a confusion matrix
- The confusion matrix shows how many from each manually annotated class were automatically classified in each of the classes

	1	2	3	4	5
1	4	5	2	0	0
2	4	4	2	1	1
3	2	4	2	2	4
4	1	1	2	2	4
5	0	0	1	2	5

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
- Divide the corpus into 5 sub-corpora; train on ABCD and test on E; train on ABCE and test on D; etc.; average the results
- The 5-fold averaged result is more meaningful than the result obtained by training on 80% of the corpus and testing on the other 20% once.
- In GATE, you can't use the Corpus QA tool on the result, but you can get a detailed statistical report at the end, including P, R, & F1 for each class

Want to give it a go?

- You can try out some ML yourself in GATE by downloading the material from Modules 11 and 12 of the GATE training course

  https://gate.ac.uk/wiki/TrainingCourseJune2012/

- You'll also find more about the evaluation tools in GATE in Module 2 of the GATE training course

Rule-based techniques

Rule-based techniques

- These rely primarily on sentiment dictionaries, plus some rules to do things like attach sentiments to targets, or modify the sentiment scores
- Examples include:
  - analysis of political tweets (Maynard and Funk, 2011)
  - analysis of opinions expressed about political events and rock festivals in social media (Maynard, Bontcheva and Rout, 2012)
  - SO-CAL (Taboada et al, 2011) for detecting positive and negative sentiment of ePinions reviews on the web.

Case study 2: Rule-based Opinion Mining from Political Tweets

Processing political tweets

- Application to associate people with their political leanings, based on pre-election tweets
  – e.g. “Had the pleasure of formally proposing Stuart King as Labour candidate for Putney”
- 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
- John may be equally in favour of several different parties, not just Labour, but hates the Conservatives above all else

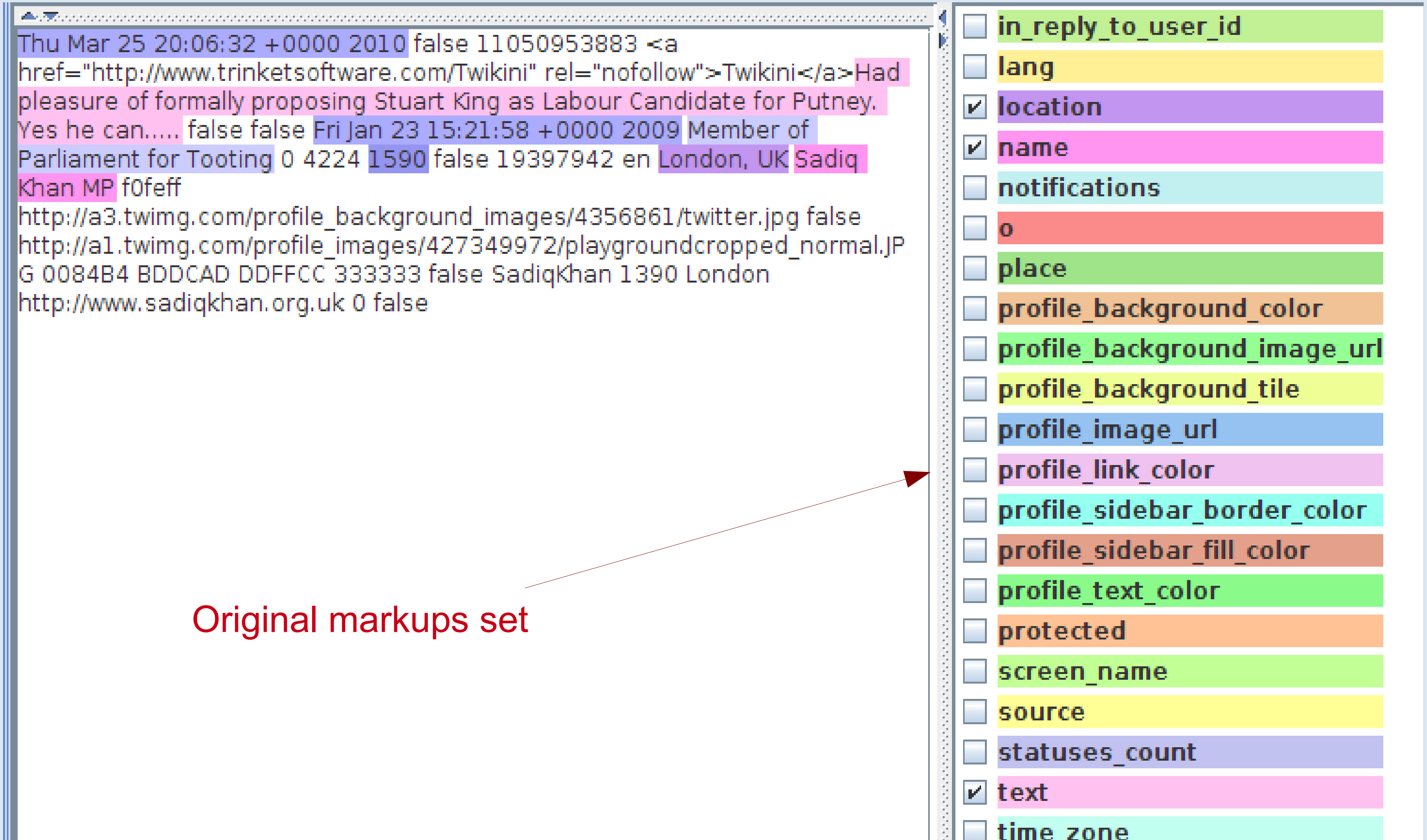
Creating a corpus

- First step is to create a corpus of tweets
- Use 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.)
- Collect tweets in json format and then convert these to xml using JSON-Lib library
- 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

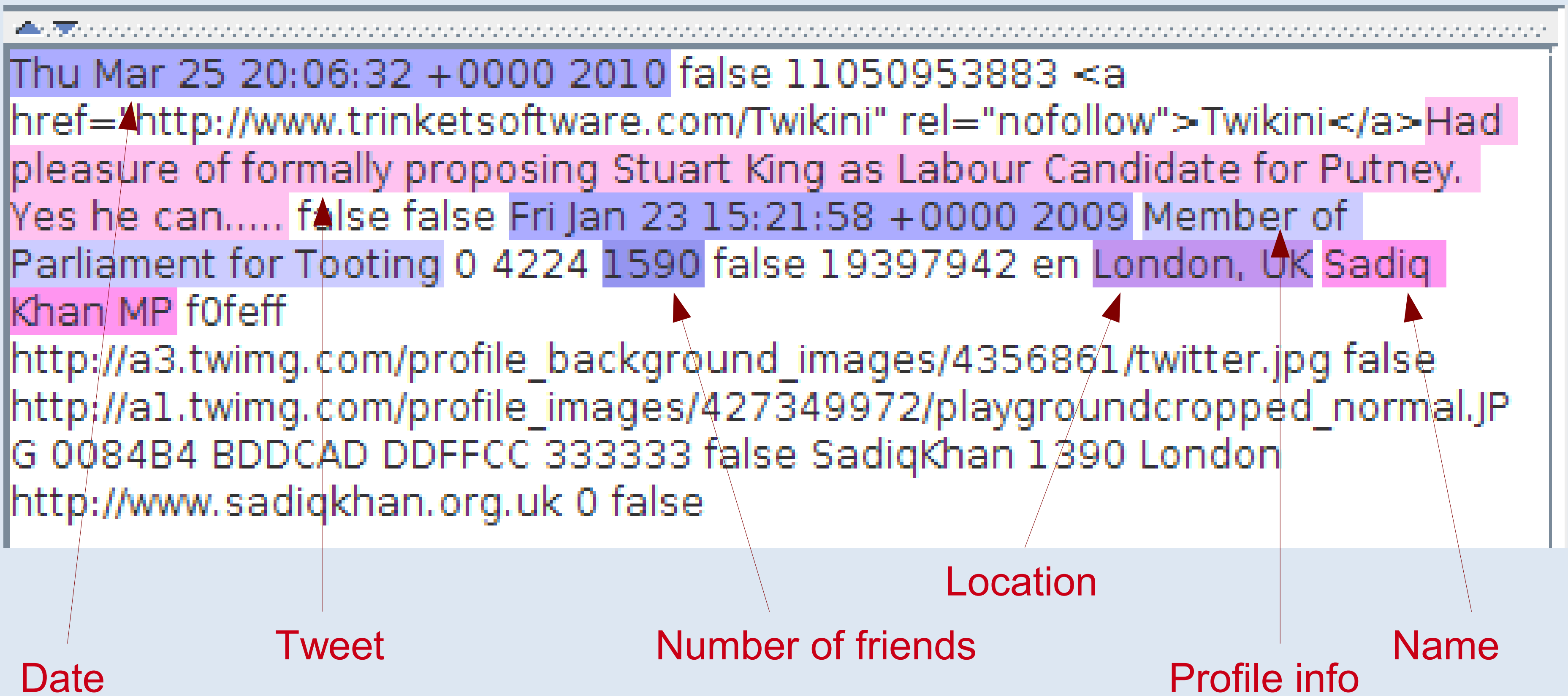
Corpus Size

- 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



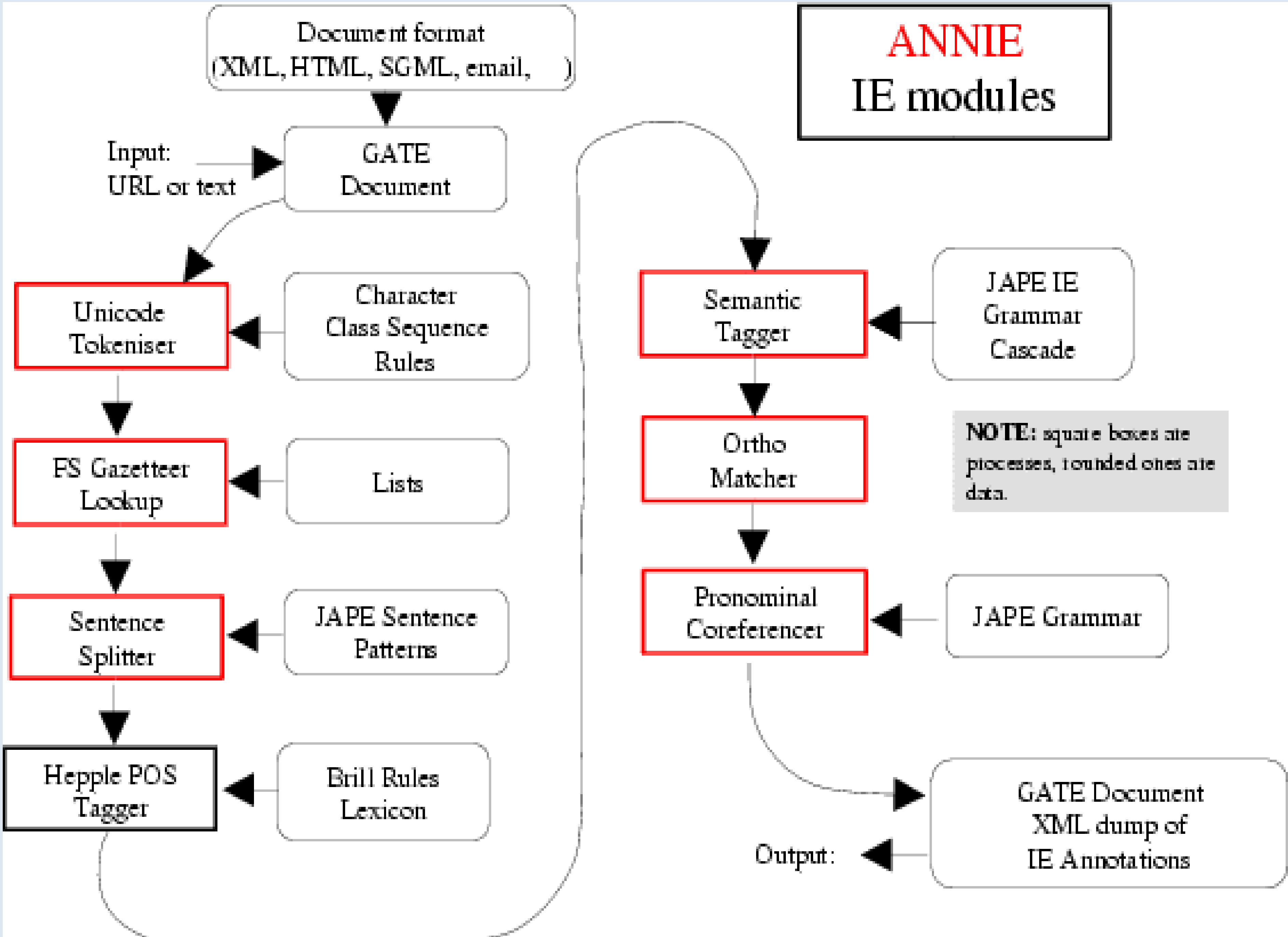
Metadata



Thu Mar 25 20:06:32 +0000 2010 false 11050953883 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.JPG 0084B4 BDDCAD DDFFCC 333333 false SadiqKhan 1390 London http://www.sadiqkhan.org.uk 0 false
Location
Tweet
Number of friends
Name
Date
Profile info

Linguistic pre-processing

- Use standard set of pre-processing resources in GATE to identify tokens, sentences, POS tags etc., and also to perform NE recognition.
- Slightly adapted the standard ANNIE application



ANNIE
IE modules
Document format
(XML, HTML, SGML, email, …)
Input:
URL or text
GATE
Document
Unicode
Tokeniser
Character
Class Sequence
Rules
FS Gazetteer
Lookup
Lists
Sentence
Splitter
JAPE Sentence
Patterns
Hepple POS
Tagger
Brill Rules
Lexicon
Semantic
Tagger
JAPE IE
Grammar
Cascade
Ortho
Matcher
NOTE: square boxes are processes, rounded ones are data.
Pronominal
Coreferencer
JAPE Grammar
GATE Document
XML dump of
IE Annotations
Output:

Gazetteers

- We create 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

A negative sentiment list

Examples of phrases following the word “go”:

- down the pan
- down the drain
- to the dogs
- downhill
- pear-shaped

A positive sentiment list

- awesome category=adjective score=0.5
- beaming category=adjective score=0.5
- becharm category=verb score=0.5
- belonging category=noun score=0.5
- benefic category=adjective score=0.5
- benevolently category=adverb score=0.5
- caring category=noun score=0.5
- charitable category=adjective score=0.5
- charm category=verb score=0.5

Grammar rules: creating preliminary annotations

- Identify questions or doubtful statements as opposed to "factual" statements in tweets, e.g. look for question marks

Wont Unite's victory be beneficial to Labour?

- Create temporary Sentiment annotations if a Sentiment 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."

Rule to match POS tag

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

```
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"}
```

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

  John's voting Labour.

- 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

  John says he's going to vote Labour.

- 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.

  Vote for Labour. Harry Potter would.

- If we want, we can grab the full details of the twitterer (author) from the metadata.

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 ANNE 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
- More info in the accompanying hands-on material

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

Name	Type	Required	Value
controller	CorpusController	✓	twitter app
inputASName	String		Original markups
segmentAnnotationType	String	✓	text

Case study 3: Opinion Mining in the Arcomem application

Arcomem

- Arcomem is an EU project about preserving community memories and retrieving interesting information from social media
- Aims to answer questions such as:
  - What are the opinions on crucial social events and the key people involved?
  - How are these opinions distributed in relation to demographic user data?
  - How have these opinions evolved?
  - Who are the opinion leaders?
  - What is their impact and influence?

Arcomem Applications

- Develop an initial application for opinion mining from social media in English and German
- Extended the political opinions application to more generic analysis about any kind of entity or event, in 2 domains
  - Greek financial crisis
  - Rock am Ring (German rock festival)
- Uses a variety of social media including twitter, facebook and forum posts
- Based on entity and event extraction, and uses a rule-based approach

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

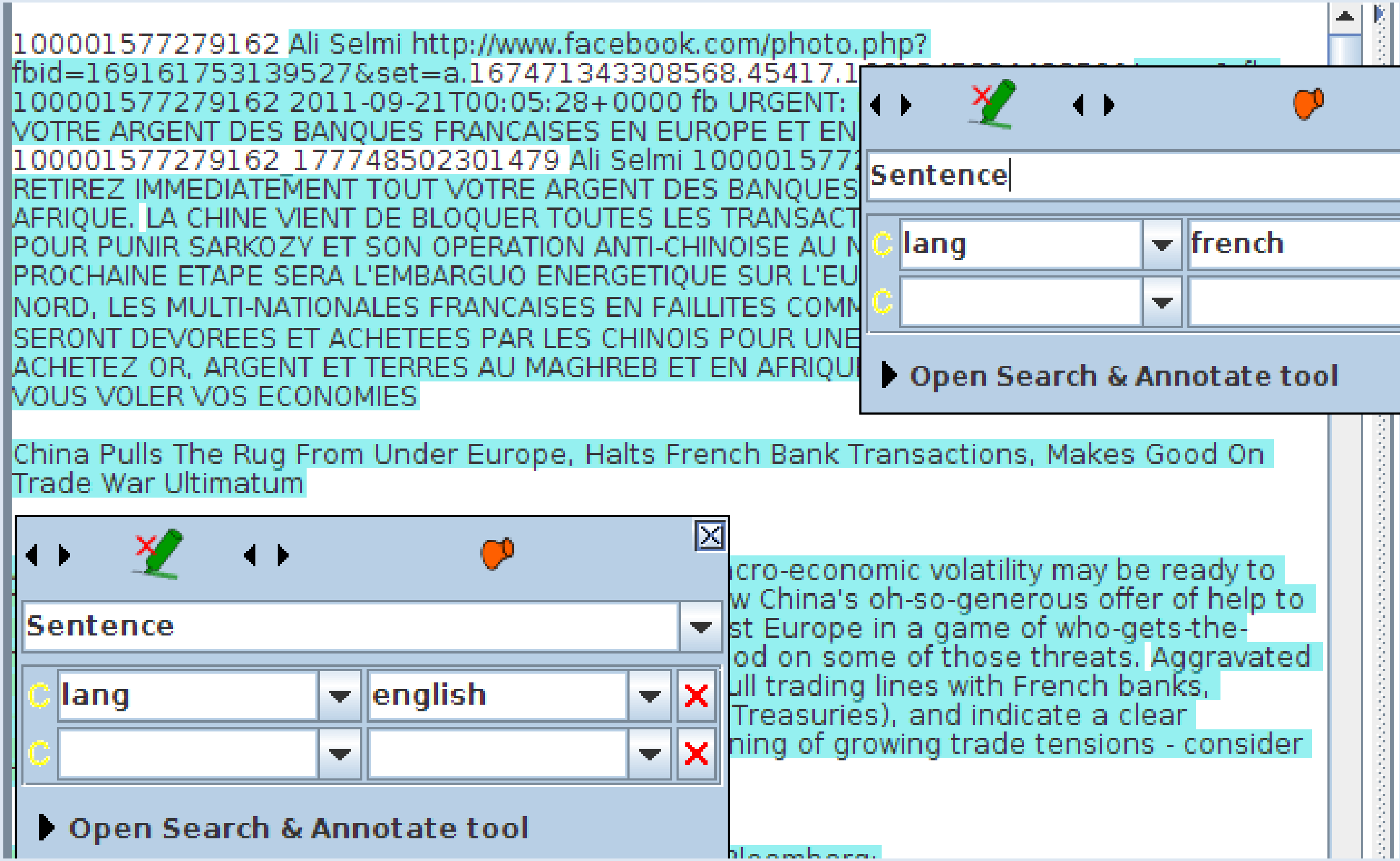
GATE Application

- Structural pre-processing, specific to social media types (such as separating the actual content of the tweet from the metadata)
- Linguistic pre-processing (including language detection), NE, term and event recognition
- Additional targeted gazetteer lookup
- JAPE grammars
- Aggregation of opinions
- Dynamics

Linguistic pre-processing

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

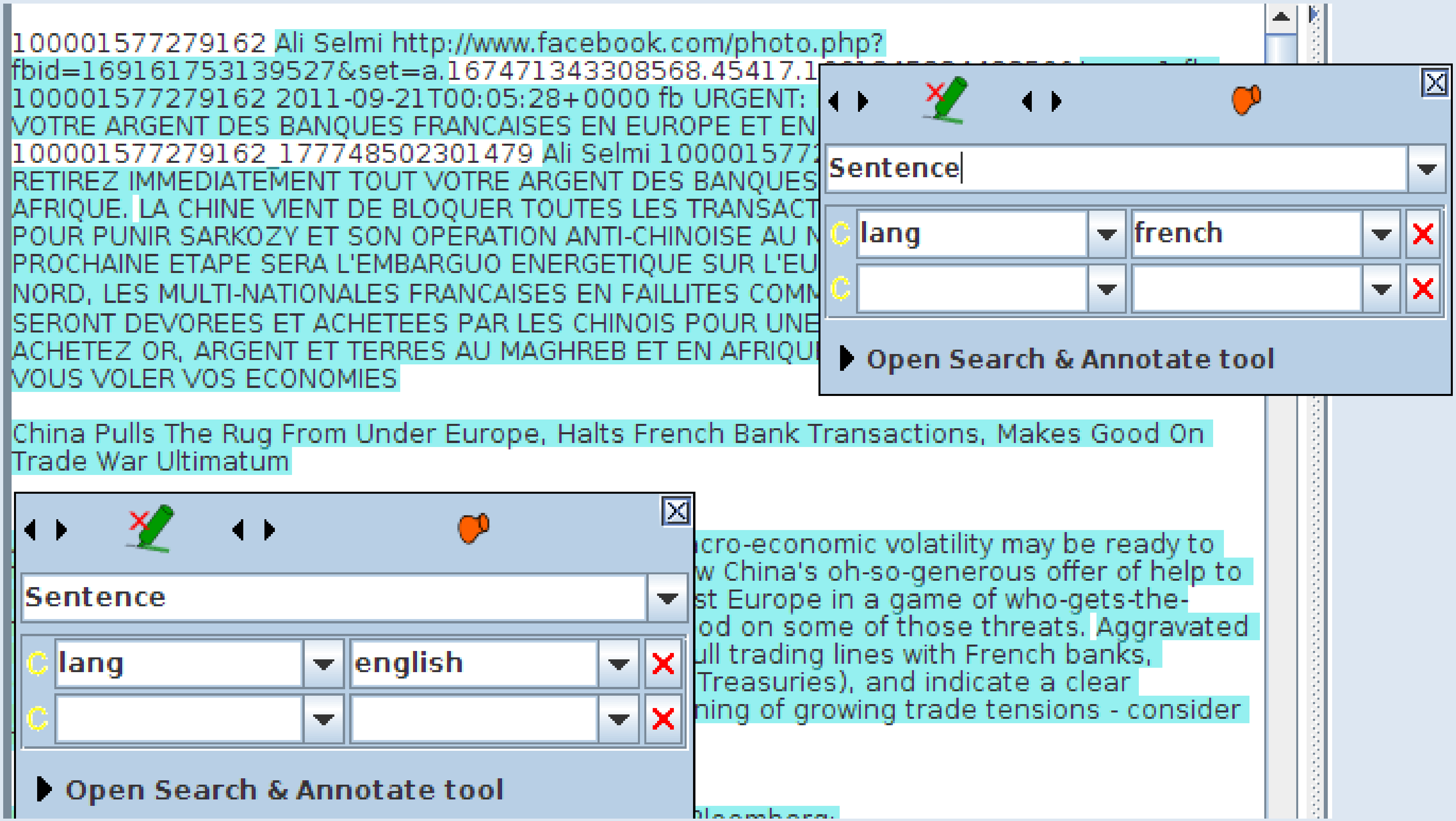
Language ID with TextCat



Linguistic pre-processing

- Language identification (per sentence) using TextCat
- Standard tokenisation, POS tagging etc using GATE
- NE and Term recognition using modified versions of ANNIE and TermRaider
- 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
- **RDF Generation**: create the relevant RDF-XML for the annotations according to the data model (so they can be used by other components)

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

  :-)

  The film was awesome --> *The film was **** amazing*.

  The film was awful --> *The film was **** awful..*

- Swear words on their own are classified as negative, however.

  Damed politicians and their lies.

  RIP Fergie? It's SIR Alex Ferguson to you, Carlos, you runt.

Evaluation

- Very hard to measure opinion polarity beyond positive / negative / neutral unless you have a product review corpus
- We did some evaluation comparing performance on political tweets, financial crisis facebook posts and financial crisis tweets
- Some interesting observatons about difficulty level
- Surprisingly, performance was better on tweets than facebook posts, though the tweets were mainly written in good English
- Detecting political affiliation was much easier than general opinions, especially wrt target assignment

Comparison of Opinion Finding in Different Tasks

Corpus	Sentiment detection	Polarity detection	Target assignment
Political Tweets	78%	79%	97.9%
Financial Crisis Facebook	55%	81.8%	32.7%
Financial Crisis Tweets	90%	93.8%	66.7%

Using Machine Learning for the Arcomem task

- If we can collect enough manually annotated training data, we can also use an ML approach for this task
- Similar to that presented earlier for the product reviews, but modified to take into account what we have subsequently learned and the differences in the data.
- Each product review had an opinion from 1 to 5 stars
- In Arcomem we classify sentences (the ML *instances)*, many of which do not contain opinions
- So the ML *classes* will be *positive*, *neutral*, *negative*, and *none* (contains no opinion, different from a *neutral* opinion)

Using Machine Learning for the Arcomem task

- We now know that PAUM is much faster than SVM but typically just as good for NLP tasks, so we will use PAUM instead
- We'll need to deal with the special issues of social media text (more on this later)
- For the ML *attributes*, we will use n-grams of tokens or lemmata
  - With the product reviews, n-grams with n>2 did not improve accuracy but slowed the ML down
  - But it's worth trying 3-grams just in case they help with the smaller instances

Should we use Named Entity info?

- Also worth trying other annotations such as named entities
- But these might exaggerate the effect of biased training data (this might not be a problem, but it's worth bearing in mind)
- For example, if most people who mention “Venus Williams” in the training data like her (or her dresses), we are training the ML model to expect positive opinions for that Person annotation; the real data might or might not match

Training on tweets

- You can use hashtags as a source of classes
  - Example: collect a set of tweets with the **#angry** tag, and a set without it, and delete from the second set any tweets that look angry
  - Remove the **#angry** tag from the text in the first set (so you're not just training the ML to spot the tag)
  - You now have a corpus of manually annotated angry/non-angry data
- This approach can work well, but if you have huge datasets, you may not be able to do the manual deletions
- Experimenting with **#sarcasm** is interesting (more on this later)

Dealing with social media challenges

Linguistic issues

- What kinds of linguistic problems do we need to overcome?
  - Short sentences (problems for parsers etc)
  - Use of incorrect English
  - Negatives
  - Conditional statements
  - Use of slang/swear words
  - Use of irony/sarcasm
  - Ambiguity

Short sentences, e.g. tweets

- Social media, and especially tweets, can be problematic because sentences are very short and/or incomplete
- Typically, linguistic pre-processing tools such as POS taggers and parsers do badly on such texts
- Even basic tools like language identification can have problems
- The best solution is to try not to rely too heavily on these tools
  – Does it matter if we get the wrong language for a sentence?
  – Do we actually need full parsing?
  – Can we use other clues when POS tags may be incorrect?

Dealing with incorrect English

- Frequent problem in any NLP task involving social media
- Incorrect capitalisation, spelling, grammar, made-up words (eg swear words, infixes)
- Backoff strategies include
  - normalisation
  - using more flexible gazetteer matching
  - using case-insensitive resources (but be careful)
  - avoiding full parsing and using shallow techniques
  - using very general grammar rules
  - adding specialised gazetteer entries for common mis-spellings, or using co-reference techniques

Tokenisation

- Splitting a text into its constituent parts
- Plenty of “unusual”, but very important tokens in social media:
  - @Apple – mentions of company/brand/person names
  - #fail, #SteveJobs – hashtags expressing sentiment, person or company names
  - :-(, :-), :-P – emoticons (punctuation and optionally letters)
  - URLs
- Tokenisation key for entity recognition and opinion mining
- A study of 1.1 million tweets: 26% of English tweets have a URL, 16.6% - a hashtag, and 54.8% - a user name mention [Carter, 2013].

Example

#WiredBizCon #nike vp said when @Apple saw what http://nikeplus.com did, #SteveJobs was like wow I didn't expect this at all.

- Tokenising on white space doesn't work that well: Nike and Apple are company names, but if we have tokens such as #nike and @Apple, this will make the entity recognition harder, as it will need to look at sub-token level
- Tokenising on white space and punctuation characters doesn't work well either: URLs get separated (http, nikeplus), as are emoticons and email addresses

The GATE Twitter Tokeniser

- Treat RTs, emoticons, and URLs as 1 token each
- #nike is two tokens (# and nike) plus a separate annotation HashTag covering both. Same for @mentions
- Capitalisation is preserved, but an orthography feature is added: all caps, lowercase, mixCase
- Date and phone number normalisation, lowercasing, and other such cases are optionally done later in separate modules
- Consequently, tokenisation is faster and more generic

De-duplication and Spam Removal

- Approach from [Choudhury & Breslin, #MSM2011]:
- Remove as duplicates/spam:
  – Messages with only hashtags (and optional URL)
  – Similar content, different user names and with the same timestamp are considered to be a case of multiple accounts
  – Same account, identical content are considered to be duplicate tweets
  – Same account, same content at multiple times are considered as spam tweets

Language Detection

- There are many language detection systems readily available
- The main challenges on tweets/facebook status updates:
  – the short number of tokens (10 tokens/tweet on average)
  – the noisy nature of the words (abbreviations, misspellings).
- Due to the length of the text, we can make the assumption that one tweet is written in only one language
- Most language detection tools work by building n-gram language models for each language and then assigning the text to the most probable language from the trained model.

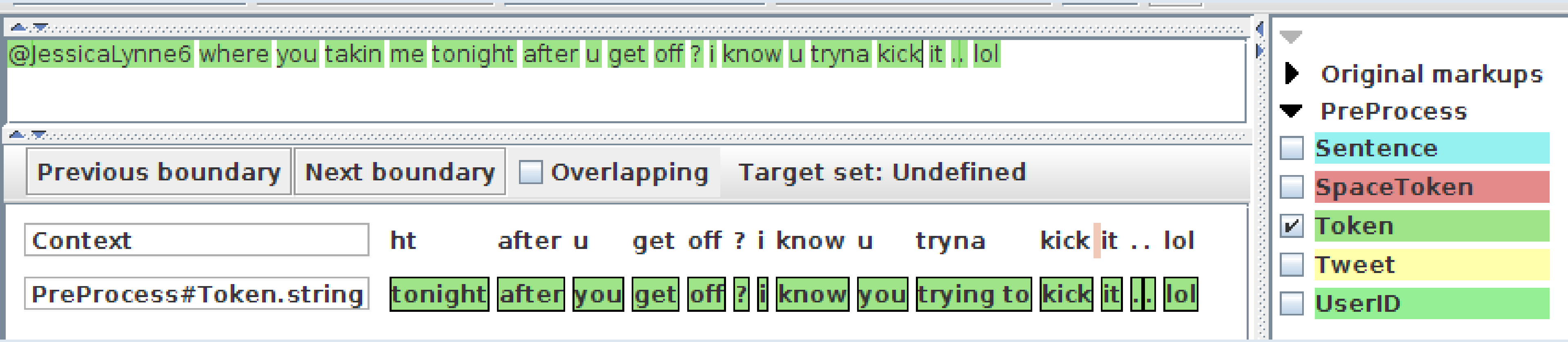
Language Detection for Social Media

- Compare language detection methods [Lui and Baldwin, 2011]
- Best results with 1-nearest-neighbour (1NN) model
  - a test document is classified based on the language of the closest training document, as determined by the cosine similarity metric
  - Character bigrams or trigrams
- We have reimplemented their best method in Java https://github.com/sinjax/trendminer-java/tree/master/text/nlp/src/main/java/org/openimaj/text/nlp
- Comes pre-trained on 97 languages and very fast

Normalisation

- “RT @Bthompson WRITEZ: @libbyabrego honored?! Everybody knows the libster is nice with it...lol...(thankkkks a bunch;))”
- OMG! I’m so guilty!!! Sprained biibii’s leg! ARGHHHHHH!!!!!!
- Similar to SMS normalisation
- For some later components to work well (POS tagger, parser), it is necessary to produce a normalised version of each token
- BUT uppercasing, and letter and exclamation mark repetition often convey strong sentiment
- Therefore some choose not to normalise, while others keep both versions of the tokens

A normalised example



- Normaliser currently based on spelling correction and some lists of common abbreviations
- Outstanding issues:
  - Insert new Token annotations, so easier to POS tag, etc? For example: “trying to” now 1 annotation
  - Some abbreviations which span token boundaries (e.g. gr8, do n’t) are not yet handled
  - Capitalisation and punctuation normalisation

NER in Tweets

- Performance of the Stanford NER drops to 48% [Liu et al, 2011] or even 29% on another tweet corpus [Ritter et al, 2011]
- Pre-processing used:
  - Stop words, user names, and links are removed
  - Specially adapted/trained POS tagger [Ritter et al, 2011]
  - NP Chunker adapted to tweets [Ritter et al, 2011]
  - Capitalisation information [Ritter et al, 2011]
  - Syntactic normalisation [Doerhmann, 2011]
  - Gazetteers derived from Freebase [Ritter et al, 2011]

More flexible matching techniques

- In GATE, as well as the standard gazetteers, we have options for modified versions which allow for more flexible matching
- BWP Gazetteer: uses Levenshein edit distance for approximate string matching
- Extended Gazetteer: has a number of parameters for matching prefixes, suffixes, initial capitalisation and so on

Extended Gazetteer

- Part of the StringAnnotation plugin in GATE
- Has the following additional characteristics:
  - Gives more control over which characters are considered to belong to words and non-word characters
  - Enables matching when an initial letter of a word is uppercase
  - matching of prefixes and suffixes
  - case-insensitive matching also deals with cases (such as German "ß" which maps to "SS")

Case-Insensitive matching

- This would seem the ideal solution, especially for gazetteer lookup, when people don't use case information as expected
- However, setting all PRs to be case-insensitive can have undesired consequences
  - POS tagging becomes unreliable (e.g. “May” vs “may”)
  - Back-off strategies may fail, e.g. unknown words beginning with a capital letter are normally assumed to be proper nouns
  - Gazetteer entries quickly become ambiguous (e.g. many place names and first names are ambiguous with common words)
- Solutions include selective use of case insensitivity, removal of ambiguous terms from lists, additional verification (e.g. use of coreference)

Resolving entity ambiguity

- We can resolve entity reference ambiguities with disambiguation techniques / linking to URI

  A plane just crashed in Paris. Two hundred French dead.

  – Paris (France), Paris (Hilton),or Paris (Texas)?
  – Match NEs in the text and assign a URI from all DBpedia matching instances
  – For ambiguous instances, calculate the disambiguation score (weighted sum of 3 metrics)
  – Select the URI with the highest score
- Try the demo at http://demos.gate.ac.uk/trendminer/obie/

Calculating disambiguation score

- Score based on
  - String similarity (edit distance between text and URI label strings)
  - Structural similarity (relation between another NE in the sentence?)

(Paris capitalOf France)

(ParisHilton bornIn NewYorkCity)

  - Contextual similarity (random indexing: probability that two words appear with a similar set of other word)
- See [Damljanovic12] for more details.

Evaluation

- How can we evaluate opinion mining performance?
- What kind of results can we expect to get?
- What problems typically occur with evaluation?
- How can we compare existing tools and methods?

Comparing different opinion mining tools

- How do you compare different opinion mining tools, when there are so many out there and they all report different kinds of results?
- It is generally accepted that tools will be 50%-70% “accurate” out-of-the box.
- But what does this really mean?
- The following 4 pieces of advice are inspired by (and adapted from) a recent article by Seth Grimes

  http://www.socialmediaexplorer.com/social-media-marketing/social-media-sentiment-competing-on-accuracy/

1. Don't compare apples with oranges

- Not all tools do the same thing, even if they look the same
- Document-level vs topic-level sentiment
- One tool might be good at getting the overall sentiment of a tweet right, but rubbish at finding the sentiment about a particular entity
- e.g. the following tweet is classed as being negative about the Olympics:

skytrain seems to be having problems frequently lately. hope cause is upgraded and they work the kinks out before olympics.

- The tweet is (correctly) negative overall but not specifically about the Olympics

2. Use the same measurement scale

- Positive/negative/neutral vs scalar measurement (-5 to +5)
- Valence vs mood/orientation (e.g. happy, sad, angry, frustrated)
- Is reasonable emotion classification more useful to you than fantastic valence?
- How will you actually make use of the opinions generated to e.g. make decisions?

3. How is accuracy defined?

- NLP tools often use Precision, Recall and F-measure to determine accuracy
- But most opinion mining tools are only measured in terms of accuracy (Precision)
- How important is Recall?
- How important is the tradeoff between Precision and Recall?
- What about *contextual* relevance that incorporates timeliness, influence, activities, and lots of other still-fuzzy *social* notions?
- How trustworthy / important are the opinions? Sentiment from a valued customer may be more important than a one-time buyer

4. What's the impact of errors?

- Not all inaccuracies have the same impact
- If you're looking at aggregate statistics, a negative rating of a positive opinion has more impact than a neutral rating of a positive opinion
- How do neutral opinions affect aggregation? Are they considered? Should they be?
- In other cases, finding any kind of sentiment (whether with correct polarity or not) might be more important than wrongly detecting no sentiment and missing important information

Creating a gold standard

- Typically, we annotate a gold standard corpus manually and then compare the system results against that
- But have you ever tried doing manual annotation of tweets?
- It's harder than it looks...
- You have to be very clear what you want to annotate
- You have to understand what the author intended
- You need to decide how lenient you'll be
- You may need to decide if getting something right for the wrong reason is still OK

Creating a gold standard

- Typically, we annotate a gold standard corpus manually and then compare the system results against that
- But have you ever tried doing manual annotation of tweets?
- It's harder than it looks...
- You have to be very clear what you want to annotate
- You have to understand what the author intended
- You need to decide how lenient you'll be
- You may need to decide if getting something right for the wrong reason is still OK

Positive or negative tweets?

RT @ssssab: Mariano: she used to be a very nice girl, before she discovered macdonalds

There was just a fire at work. Today is looking up.

Yesterday my son forgot his jacket at school. Today he remembered to bring home the jacket, but forgot his lunchbox.

I find myself sobbing at John Le Mesurier's beauty of soul. Documentary about him on BBC iPlayer

Opinionated or not?

The European sovereign debt crisis that's spread from Greece to Italy and is roiling the region's banks now has another potential victim: energy policy.

Labour got less this time than John Major did in 1997.

EUROPEAN LEADERSHIP - where is it?

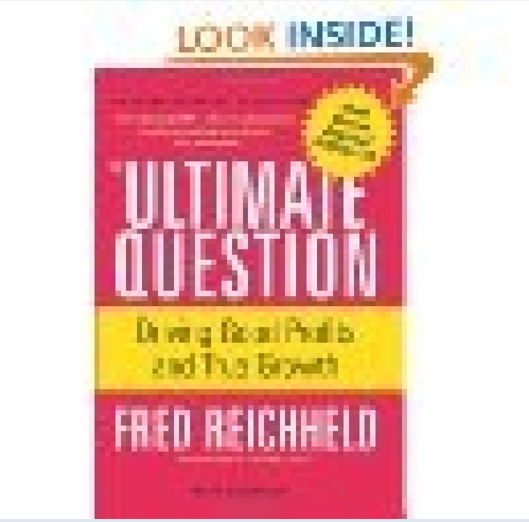
Other challenges of social media

- **Strongly temporal and dynamic**: temporal information (e.g. post timestamp) can be combined with opinion mining, to examine the volatility of attitudes towards topics over time (e.g. gay marriage).
- **Exploiting social context**: (Who is the user connected to? How frequently they interact). Derive automatically semantic models of social networks, measure user authority, cluster similar users into groups, as well as model trust and strength of connection
- **Implicit information about the user**: Research on recognising gender, location, and age of Twitter users. Helpful for generating opinion summaries by user demographics

Looking into the future

- Typically, opinion mining looks at social media content to analyse people's explicit opinions about a product or service
- This backwards-looking approach often aims primarily at dealing with problems, e.g. unflattering comments
- A forwards-looking approach aims at looking ahead to understanding potential new needs from consumers
- This is not just about looking at specific comments, e.g. "the product would be better if it had longer battery life", but also about detecting non-specific sentiment
- This is achieved by understanding people's needs and interests in a more general way, e.g. drawing conclusions from their opinions about other products, services and interests.

The Ultimate Question



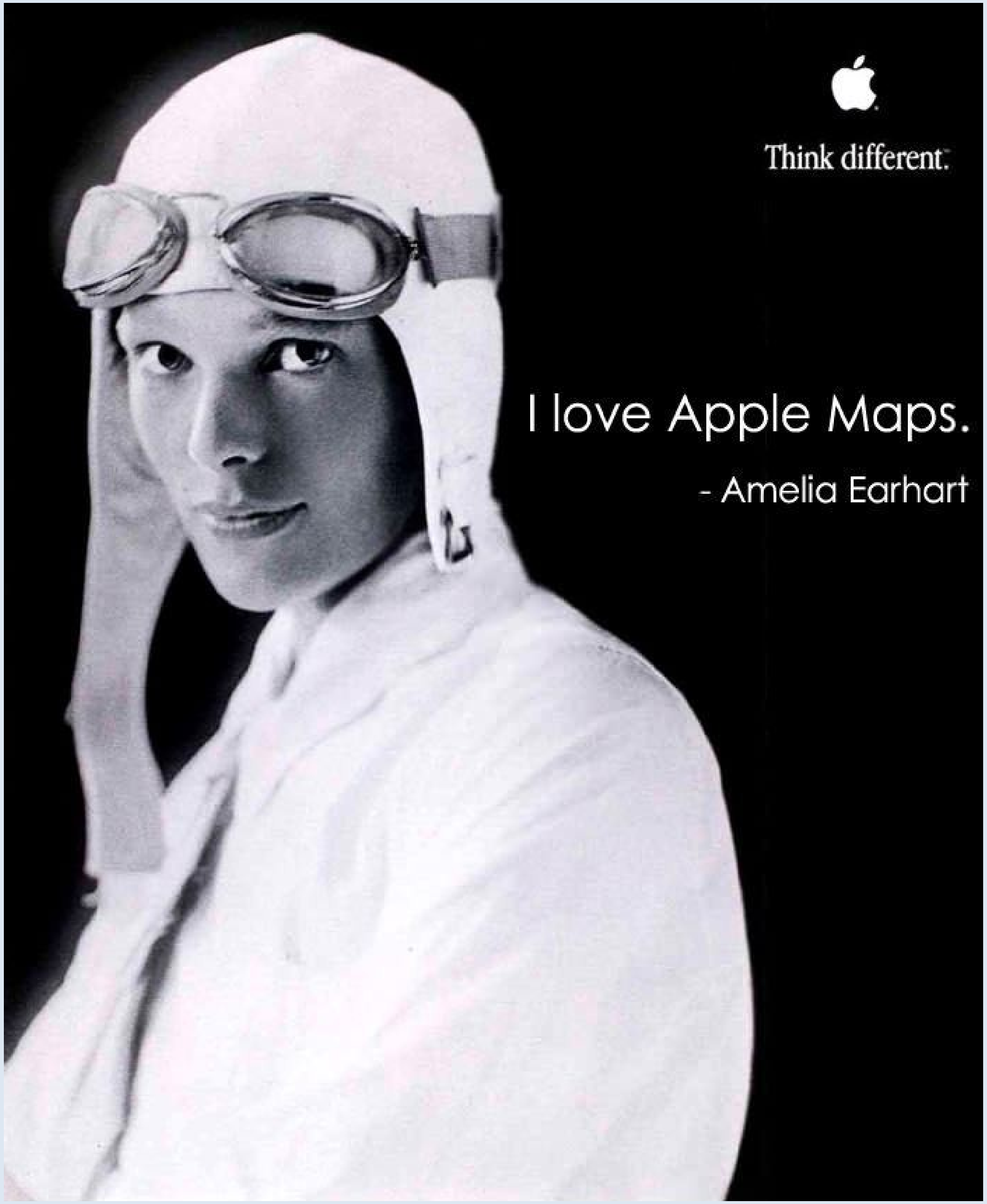
- The book "The Ultimate Question" recently ranked #1 on the Wall Street Journal's Business Best-Sellers
- List and #1 on USA TODAY's Money Best-Sellers List.
- It's all about whether a consumer likes a brand enough to recommend it - this is the key to a company's performance.
- General sentiment detection isn't precise enough to answer this kind of question, because all kinds of “like” are treated equally
- Growing need for sentiment analysis that can get to very fine levels of detail, while keeping up with the enormous (and constantly increasing) volume of social media.

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?

Take-home message

- Opinion mining is **hard** and therefore **error-prone** (despite what vendors will tell you about how great their product is)
- There are many types of sentiment analysis, and many different uses, each requiring a different solution
- It's very unlikely that an off-the-shelf tool will do exactly what you want, and even if it does, performance may be low
- Opinion mining tools need to be **customised** to the task and domain
- Anything that involves processing **social media** (especially messy stuff like Facebook posts and Twitter) is even harder, and likely to have lower performance
- For tasks that mainly look at aggregated data, this isn't such an issue, but for getting specific sentiment on individual posts/reviews etc, this is more problematic


Think different.
I love Apple Maps.
- Amelia Earhart

More information

- GATE http://gate.ac.uk (general info, download, tutorials, demos, references etc)
- The EU-funded ARCOMEM and TrendMiner projects are dealing with lots of issues about opinion and trend mining from social media, and use GATE for this.
  - http://www.arcomem.eu
  - http://www.trendminer-project.eu/
- Related tutorials
  - Module 12 of the annual GATE training course: “Opinion Mining” (2012 version available from http://gate.ac.uk/wiki/TrainingCourseJune2012/
  - Module 14 of the annual GATE training course: “GATE for social media mining”

More information on Replab

- Replab 2012 overview http://www.limosine-project.eu/events/replab2012
- Overview of RepLab 2012: Evaluating Online Reputation Management Systems http://www.limosine-project.eu/sites/default/files/replab2012overviewv2.pdf
- M. A. Greenwood, N. Aswani, K. Bontcheva: Reputation Profiling with GATE. CLEF (Online Working Notes/Labs/Workshop). 2012. http://gate.ac.uk/sale/replab2012-clef/GATE-RepLab2012.pdf

Some GATE-related opinion mining papers

(available from http://gate.ac.uk/gate/doc/papers.html)

- D. Maynard and K. Bontcheva and D. Rout. Challenges in developing opinion mining tools for social media. In Proceedings of @NLP can u tag #usergeneratedcontent?! Workshop at LREC 2012, May 2012, Istanbul, Turkey.
- M. A. Greenwood, N. Aswani, K. Bontcheva: Reputation Profiling with GATE. CLEF (Online Working Notes/Labs/Workshop). 2012.
- D. Maynard and A. Funk. Automatic detection of political opinions in tweets. In Raúl García-Castro, Dieter Fensel and Grigoris Antoniou (eds.) The Semantic Web: ESWC 2011 Selected Workshop Papers, Lecture Notes in Computer Science, Springer, 2011.
- H.Saggion, A.Funk: Extracting Opinions and Facts for Business Intelligence. Revue des Nouvelles Technologies de l'Information (RNTI), no. E-17 pp119-146; November 2009.
- Adam Funk, Yaoyong Li, Horacio Saggion, Kalina Bontcheva and Christian Leibold: Opinion Analysis for Business Intelligence Applications. In First International Workshop on Ontology-supported Business Intelligence (OBI2008) at the 7th International Semantic Web Conference (ISWC), Karlsruhe, Germany, October 2008.
- D. Damljanovic and K. Bontcheva: . Named Entity Disambiguation using Linked Data. Proceedings of the 9th Extended Semantic Web Conference (ESWC 2012), Heraklion, Greece, May 31-June 3, 2010. Poster session

References in this tutorial

- T. Baldwin and M. Lui. Language Identification: The Long and the Short of the Matter. In Proc. NAACL HLT ’10. http://www.aclweb.org/anthology/N10-1027.
- M. Kaufmann. Syntactic Normalization of Twitter Messages. http://www.cs.uccs.edu/~kalita/work/reu/REUFinalPapers2010/Kaufmann.pdf
- S. Choudhury and J. Breslin. Extracting Semantic Entities and Events from Sports Tweets. Proceedings of #MSM2011 Making Sense of Microposts. 2011.
- X. Liu, S. Zhang, F. Wei, M. Zhou. Recognizing Named Entities in Tweets. ACL'2011.
- A. Ritter, Mausam, Etzioni. Named entity recognition in tweets: an experimental study. EMNLP'2011.
- Doerhmann. Named Entity Extraction from the Colloquial Setting of Twitter. http://www.cs.uccs.edu/~kalita/work/reu/REU2011/FinalPapers/Doehermann.pdf
- S. Carter, W. Weerkamp, E. Tsagkias. Microblog Language Identification: Overcoming the Limitations of Short, Unedited and Idiomatic Text. Language Resources and Evaluation Journal. 2013 (Forthcoming)
- Johan Bollen, Huina Mao, Xiaojun Zeng, Twitter mood predicts the stock market, Journal of Computational Science, Volume 2, Issue 1, March 2011..

Some demos to try

- http://sentiment.christopherpotts.net/lexicon/ Get sentiment scores for single words from a variety of sentiment lexicons
- http://sentiment.christopherpotts.net/textscores/ Show how a variety of lexicons score novel texts
- http://sentiment.christopherpotts.net/classify/ Classify tweets according to various probabilistic classifier models

More information about this tutorial

- This tutorial and additional information can be found at

  http://www.dcs.shef.ac.uk/~diana/courses/sentiment-symposium2012-tutorial.html
- This includes:
  - some hands-on exercises for getting your hands dirty with opinion mining in GATE
  - accompanying sample applications and test corpora
  - information about how to get more help, training or consultancy on opinion mining and other forms of text analytics from the GATE team

Questions?