

Introduction to Text Mining

Module 3: 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...



Venus Williams causes controversy...



Opinion mining exposes these insights

Get Tips by Timeline ?



Positive
Negative
Neutral

Move slider to search
Day(s) Just Current Latest
earlier prior

Mostly from 4 days -Now

Discover Tips by Topic ?

[pie](#) [venue](#) [dress](#) [is](#) [court](#)
[williams](#) [crust](#) [tennis](#)

😊 Positive Tips ?



I see a lot of people on here bitching about **Venus Williams' dress**. I really like it. She looks really good in it.
#WTA #Tennis



Venus Williams' dress looks as if it might be worn by Hannibal Lecter to keep him from eating people. Only in a cheery color.

😞 Negative Tips ?



Watching the Australian Open and am quite disturbed by **Venus Williams' dress** which has completely unnecessary holes all around



Srsly, this **Venus Williams dress** is as bad as her backhand, which is all elbows.

😐 Neutral Tips ?



Someone took **Venus Williams' dress** and police have cordoned off the area.
#ausopen



Soooo I just saw **Venus Williams' dress** and now I want a Belgian Waffle from [@MaxBrenner](#).

Online social media sentiment apps

- Try a search of your own (e.g. “Gaddafi”) on one of these:
 - Twitter sentiment <http://twittersentiment.appspot.com/>
 - Twends: <http://twendz.waggeneredstrom.com/>
 - Twittratr: <http://twitrratr.com/>
 - SocialMention: <http://socialmention.com/>
- 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 Venus Williams, you can only search on “Venus Williams” to get hits

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



Related (sub)topics: general

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

Feature-opinion association

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

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

- **Spam opinion detection:** detect whether opinions are written by spammers
- Could involve language analysis, frequency analysis,...
- Could be either positive or negative opinions
- Generally, negative opinions are more damaging than positive ones
- Spam opinions are typically posted by the same person on multiple sites and/or threads with identical or very similar wording
- Sometimes people get paid to post “spam” opinions supporting a product, organisation or even government: typically these are easily detectable due to their similarity and frequency
- Look for **outlier** reviews (which go against the trend)

It's not just about cameras and dresses ...

- Film, theatre, books, fashion etc
 - impacts on the whole industry
 - predictions about changing society, trends etc.
- Monitoring political views
- Feedback/opinions about multimedia productions, e.g. documentaries, broadcasts etc.
- Feedback about events, e.g. conferences
- Scientific and technological monitoring, competitor surveillance etc.
- Monitoring public opinion
- Creating community memories

And it's not always as easy as it looks...



“Rubbish hotel in Madrid”

Deep sentiment analysis

- The hardest thing about getting sentiment analysis right is uncovering exactly what is being meant
- Difference between a customer saying they merely like a brand and saying that they love it.
- Sentiment has many rich and nuanced dimensions that need to be teased apart to make it insightful.
- “An old lady told me that warm Dr. Pepper is delicious”
 - Is it only nice when warm? Does the author share the opinion of the old lady?
 - Could this be a new insight for the manufacturers/advertisers?
- “I loved Dr. Pepper because of Dude.” (Dude was a character in the ad)
 - Does the author still like Dr Pepper?
 - Should the manufacturers consider bringing back the commercial?
- Classification of sentiment according to functional, insightful, emotional etc.

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 apparently 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.

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

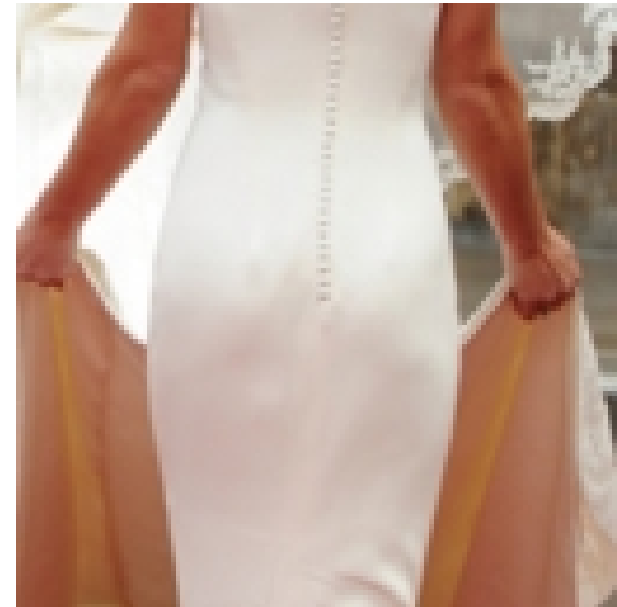
- This kind of predictive analysis is all about understanding your potential audience at a much deeper level - this can lead to improved advertising techniques such as personalised ads to different groups

Analysing and preserving opinions

- Useful to collect, store and later retrieve public opinions about events and their changes or developments over time
- One of the difficulties lies in distinguishing what is important
- Opinion mining tools can help here
- Not only can online social networks provide a snapshot of such situations, but they can actually trigger a chain of reactions and events
- Ultimately these events might lead to societal, political or administrative changes

Pippa Middleton's assets

- One of the biggest Royal Wedding stories on Social Media sites
- Her bottom has its own [twitter account](#), [facebook page](#) and [website](#).
- Pilates classes have become incredibly popular since the Royal Wedding, solely as a result of all the social media



Pippa Middleton has revealed the secret to her perfect figure - Pilates classes. <http://dlvr.it/S9Cy8>

[CutePippaFace](#): 57 minutes ago.

[Reply](#) [View Tweet](#)

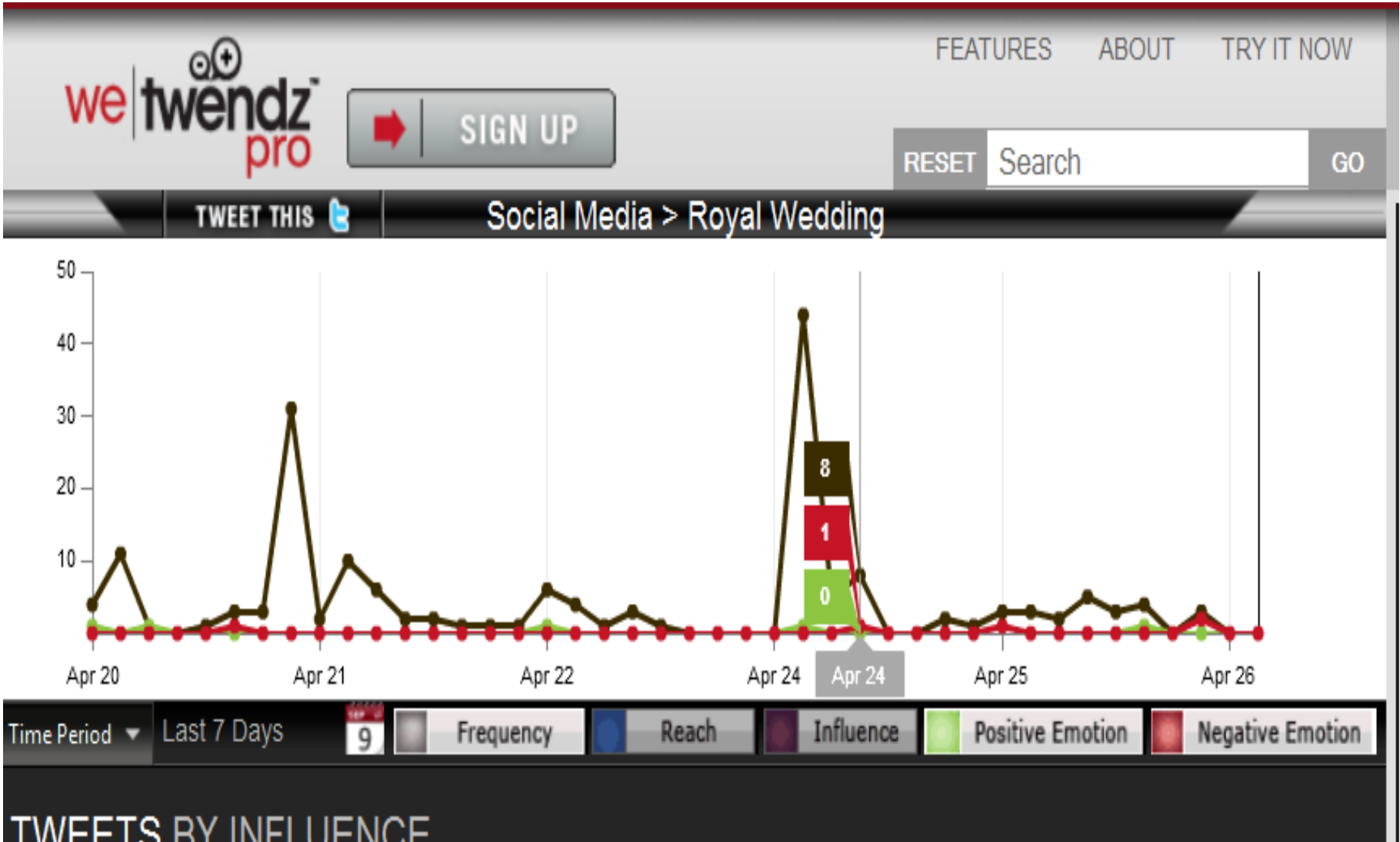
Accuracy of twitter sentiment apps

- Mine the social media sentiment apps and you'll find a huge difference of opinions about Pippa Middleton:
 - **TweetFeel**: 25% positive, 75% negative
 - **Twendz**: no results
 - **TipTop**: 42% positive, 11% negative
 - **Twitter Sentiment**: 62% positive, 38% negative
- Try searching for “Gaddafi” and you may be surprised at the results.

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



Mapping dynamics from social media: UK riots

The image shows a satellite view of a location in Enfield, UK, with an information window overlaid. The window contains the following text:

Sony Distribution Centre

Enfield

What happened? Set on fire. Completely engulfed by flames and destroyed

Time (Approx), if known 23:49

Date: Aug 8, 2011

Day (from noon to 6am) Monday

Source [Link to report, if known](#)

[Access this Fusion table data](#)

Google Imagery ©2011 TerraMetrics - [Terms of Use](#)

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
 - 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

Case study: 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 JSON-:ib 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

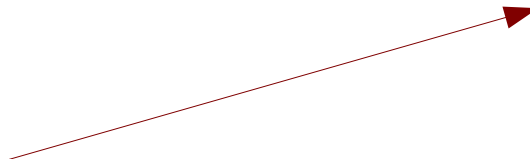
- Raw corpus contained around 5 million tweets
- Many were duplicates due to the way in which the tweets were collected
- Added a de-duplication step during the conversion of json to xml
- This reduced corpus size by 20% to around 4 million
- This still retains the retweets, however

Tweets with metadata

```
Thu Mar 25 20:06:32 +0000 2010 false 11050953883 <a href="http://www.trinketsoftware.com/Twikini" rel="nofollow">Twikini</a>Had 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 DDFCC 333333 false SadiqKhan 1390 London http://www.sadiqkhan.org.uk 0 false
```

- in_reply_to_user_id
- lang
- location
- name
- notifications
- o
- place
- profile_background_color
- profile_background_image_url
- profile_background_tile
- profile_image_url
- profile_link_color
- profile_sidebar_border_color
- profile_sidebar_fill_color
- profile_text_color
- protected
- screen_name
- source
- statuses_count
- text
- time_zone

Original markups set



Metadata

```
Thu Mar 25 20:06:32 +0000 2010 false 11050953883 <a href="http://www.trinketsoftware.com/Twikini" rel="nofollow">Twikini</a> Had 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
```

Date

Tweet

Number of friends

Location

Profile info

Name

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

Grammar rules: creating temporary 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 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 awful days of Tory rule” vs “Ah good, the entertainment is here.”*
 - *“People like her should be shot.” vs “People like her.”*

Question grammar

Phase: Preprocess

Input: Token

Options: control = appelt

Rule: Question

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

Phase: Affect

Input: AffectLookup Token

Options: control = appelt

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`
- 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`.
- Later we can grab the details of the twitterer instead of just using "author".

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





Implicit Opinion Holders

- There may not always be an explicit opinion holder
- In many cases, the author of the tweet is the opinion holder
 - *I'm also going to vote Tory. Hello new world.*
 - Here we can co-refer “I” with the person tweeting (using the metadata)
- In other cases, there is no explicit opinion holder:
 - *“Vote for Labour. Harry Potter would.”*
 - However, we can infer by this instruction that the author of the tweet shares this opinion.
- In all these cases, we add the value “author” to the feature “opinion_holder”

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
 controller	CorpusController	✓	 twitter app
 inputASName	String		Original markups
 segmentAnnotationType	String	✓	text

Demo and Hands-on: Analysing political tweets

- Load the file **politwits-datastore/politwits-1000-ds** as a GATE Lucene datastore (right click on **Datastores**, select **Open Datastore**, then **Lucene datastore** and select the file)
- Double click on the datastore name to open it
- In the datastore view, double click on a document to load it
- Open a document and look at the results
- Click on **Sentiment** to see the overall Sentiment of the tweet
- Click on **Party** and **Vote** to see intermediate annotations

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
- May help in refining rule-writing

ANNIC Demo and HandsOn

- Formulating queries
- Finding matches in the corpus
- Analysing the contexts
- Refining the queries
- Demo: <http://gate.ac.uk/demos/annic2008/Annic-only.htm>

Demo and Hands-On: Using ANNIC

- Double click on the datastore politwits-500 that you have loaded in GATE to open the datastore viewer
- Select “Lucene datastore searcher” from the datastore viewer (bottom pane)
- Try out some patterns to see what results you get, e.g. {Sentiment}
- Hint: click on the name of an annotation in the bottom right corner, to add it to the search box, or start typing in the search box to get some help with possible annotations.

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"}

How can we improve on a basic opinion mining system?

Linguistic issues

- What kinds of linguistic problems do we need to overcome?
 - Negatives
 - Conditional statements
 - Use of incorrect English
 - Use of slang/swear words
 - Use of irony/sarcasm

Finding Negatives

- What methods might we use for finding negatives?
 - List lookup
 - Verb analysis
 - Sarcasm

Conditional statements

- Linguistic information can give you a lot of clues about meaning
- We would normally expect “good battery life” to indicate a positive feature (in the absence of negation).
- Can you think of any counter-examples?

Conditional statements

- 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

Sample grammar for type 0 conditional

Input: Split VG ConditionalIndicator

Pattern: If + simple present, simple present

Rule: Conditional0

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

List Lookup

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

Demo and Hands-On: Conditionals

- Load the Lucene datastore “movie-polarity-ds” into GATE
- Select “Lucene datastore searcher” from the datastore viewer
- Search for “Conditional” and see what results you get
- Select “Conditional” with feature “type” in the annotation viewer to see the different types of conditionals

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”

Dealing with incorrect English

- Frequent problem in any NLP task involving social media
- Backoff strategies include
 - 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 misspellings, or using co-reference techniques

Dealing with irony and sarcasm

- “Pretty sure that the heavy downpour of rain in the Withington/Didsbury area just knocked out our sky aerial. Brilliant.”
- “The now abandoned HP TouchPad is officially the hottest piece of consumer electronics on Amazon.”
- “Loves being in this supah long line at the #DMV -- woo hoo”
- “AWESOME! A whole new round of bickering around a vaccine in Texas. Excellent.”
- “I'm so glad we have David Cameron to sort out our broken society. Someone who really represents the working man.”
- “Well. This is a great start to the semester..”

How do you know when someone is being ironic/sarcastic?

- Use of the #sarcasm hashtag - this can also be used to make a training set for machine learning
- Look for word combinations with opposite polarity, e.g. “rain” or “delay” plus “brilliant”
- Inclusion of world knowledge / ontologies can help

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.

Questions

1. How might we identify tweets such as the following as irrelevant in terms of political opinion?

I am working at Tory's pre - school. I substitute there sometimes 2 - 4 days a week.

2. How might we incorporate recognition of swear words to improve our opinion mining from tweets?
3. How could you use an ontology to make generalisations over tweets about political opinions?

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
- Tools for corpus analysis
- Examples of how deeper linguistic information can be useful
- Looking ahead to the future

More information

- See the following paper for details and evaluation of a more complex version of the twitter application
- D. Maynard and A. Funk. Automatic detection of political opinions in tweets. In Proceedings of MSM 2011: Making Sense of Microposts. Workshop at 8th Extended Semantic Web Conference (ESWC 2011). Heraklion, Greece. June 2011 (download [PDF](#))
- The EU-funded ARCOMEM project is dealing with lots of issues about opinion mining from social media, and uses GATE for this. See <http://www.arcomem.eu> for more details