Module 3: GATE and Social Media

3: TwitIE components

NLP Pipelines


BBC | News | Sport | Weather | Travel | Future | Autos

NEWS ASIA

Home | UK | Africa | Asia | Europe | Latin America | Mid-East | US & Canada | Business | Health | Sci/Enviro

China | India

30 April 2013 Last updated at 23:50 GMT

166 Share

Viewpoint: Are US-North Korea talks possible?

By Andrea Berger
Royal United Services Institute

While North Korean rhetoric appears to be finally tapering off, tensions on the Korean Peninsula persist. In a matter of only a few weeks, the Kim regime has unravelled two of the greatest successes of detente with its adversaries.

Pyongyang announced that it would restart production of plutonium for nuclear weapons at its Yongbyon reactor, whose cooling tower was famously demolished in 2008 as part of its commitment to the six-party talks.

North Korea says it will not talk until all UN sanctions against it have been repealed

More recently, it has withdrawn from the joint Kaesong Industrial Zone with South Korea.

These developments serve as a reminder that talks with North Korea can yield modest, but desirable outcomes like the establishment of Kaesong.

Korea crisis

China's role

Should we fear?

Text

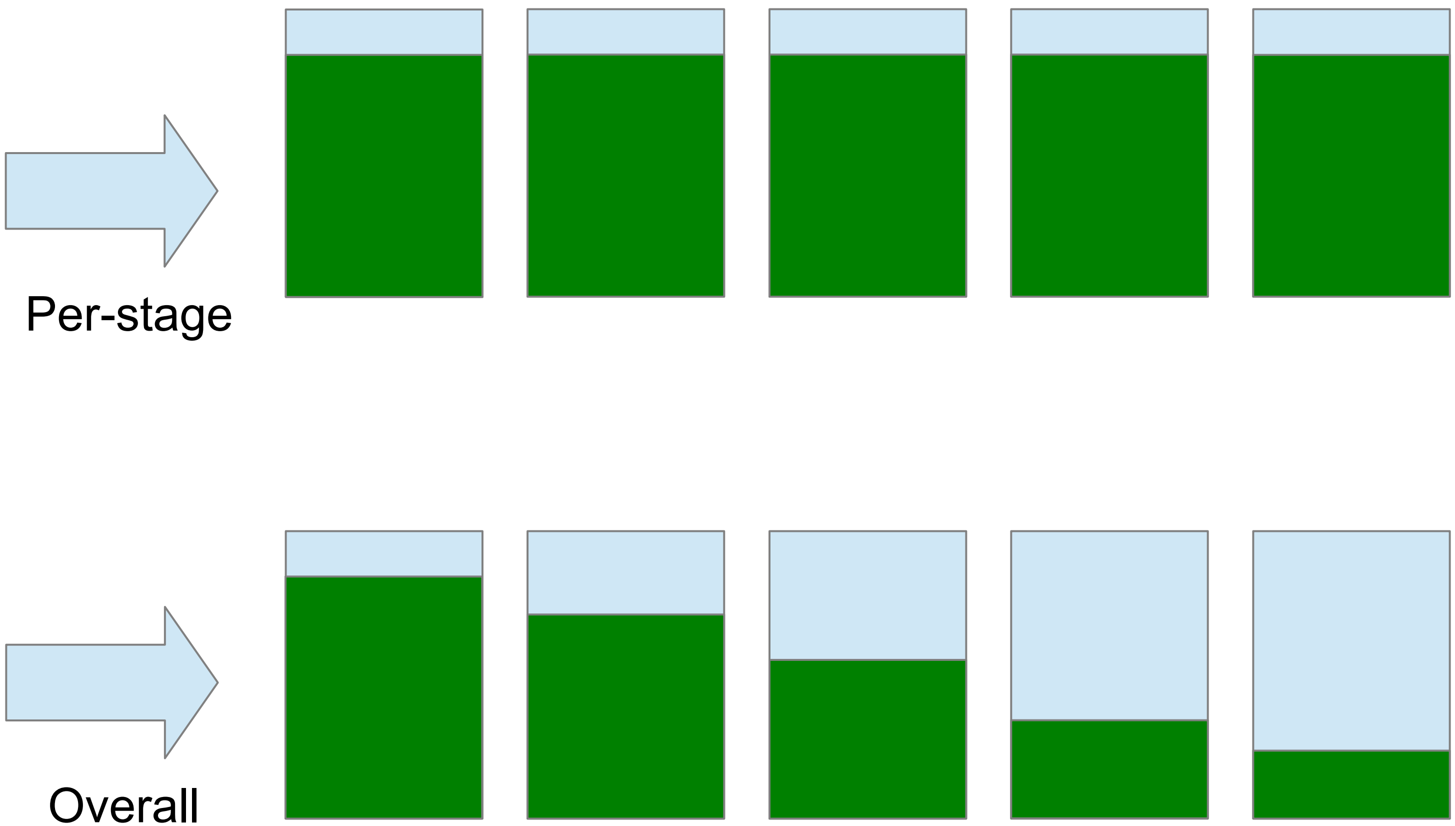
Language ID

Tokenisation

Part of speech tagging

Pipelines for tweets

- Errors have a cumulative effect



Good performance is important at each stage

Language ID: example

Task: given a text, determine which language it is in

Newswire:

The Jan. 21 show started with the unveiling of an impressive three-story castle from which Gaga emerges. The band members were in various portals, separated from each other for most of the show. For the next 2 hours and 15 minutes, Lady Gaga repeatedly stormed the moveable castle, turning it into her own gothic Barbie Dreamhouse .

Language ID: example

Task: given a text, determine which language it is in

Newswire: The Jan. 21 show started with the unveiling of an impressive three-story castle from which Gaga emerges. The band members were in various portals, separated from each other for most of the show. For the next 2 hours and 15 minutes, Lady Gaga repeatedly stormed the moveable castle, turning it into her own gothic Barbie Dreamhouse .

Twitter: LADY GAGA IS BETTER THE 5th TIME OH BABY(:

je bent Jacques cousteau niet die een nieuwe soort heeft ontdekt, het is duidelijk, ze bedekken hun gezicht. Get over it

I'm at 地铁望京站 Subway Wangjing (Beijing) http://t.co/KxHzYm00

RT @TomPIngram: VIVA LAS VEGAS 16 - NEWS #constantcontact http://t.co/VrFzZaa7

Language ID: issues

Accuracy on microblogs: 89.5% (Preotiuc-Pietro 2012)
Accuracy on formal text: 99.4% (Carter 2013)

What general problems are there in identifying language in social media?

- Switching language mid-text;
- Non-lexical tokens (URLs, hashtags, usernames, retweet/modified tweet indicators);
- Small “samples”: documents are fixed at 140 characters, and document length has a big impact on language identification;
- Dysfluencies and fragments reduce n-gram match likelihoods;
- Large (unknown) number of potential languages, some for which there will be no training data (Baldwin 2010).

Social media introduces new information

- Metadata:
  spatial information (from profile, from GPS);
  language information (default English is left on far too often).

- Emoticons:

  :) vs. ^_^

  cu vs. 88

Language ID: solutions

Carter et al. (2013) introduce semi-supervised priors to overcome short message problems:

- Author prior, using content of previous messages from the same author;
- Link prior, using text from any hyperlinks in the message;
- Mention prior, based on the author priors of other users mentioned in the message;
- Tag prior, gathering text in other messages sharing hashtags with the message;
- Conversation prior, taking content from messages in a conversation thread.

These priors individually help performance

- Author prior offers 50% error reduction, and is most helpful in five languages surveyed.
- Why? This prior will generate the most content – the others are conditional.

Language ID: solutions (2)

Combining priors leads to improved performance

- Different strategies help for different languages;
- Tried: voting, beam search, linear interpolation, beam confidence, lead confidence.
- Beam confidence (reducing prior weight when many languages close to most likely).

Tricky cases remain difficult, especially when languages mix

- Fluent multilingual posts; foreign named entities; misleading priors; language ambiguous

Language ID: solutions (3)

Carter technique can be demanding

- Data may not be available: API limits, graph changes, deleted items, changed web pages
- Processing time: retrieving required information is slow
- Privacy concerns: somewhat invasive

Lui and Baldwin (2012) use information gain-based feature selection for transductive language ID

- Goal is to develop cross-domain language identification
- In-domain language identification is significantly easier than cross-domain
- Social media text is more like a mixture of small/personal domains than its own domain

Language ID: solutions (4)

The variety of data and sparsity of features makes selection important

- LD focuses on task-relevant features using information gain
- Features with a high LD score are informative about language, without being informative about domain
- Candidate features pruned before applying LD based on term frequency

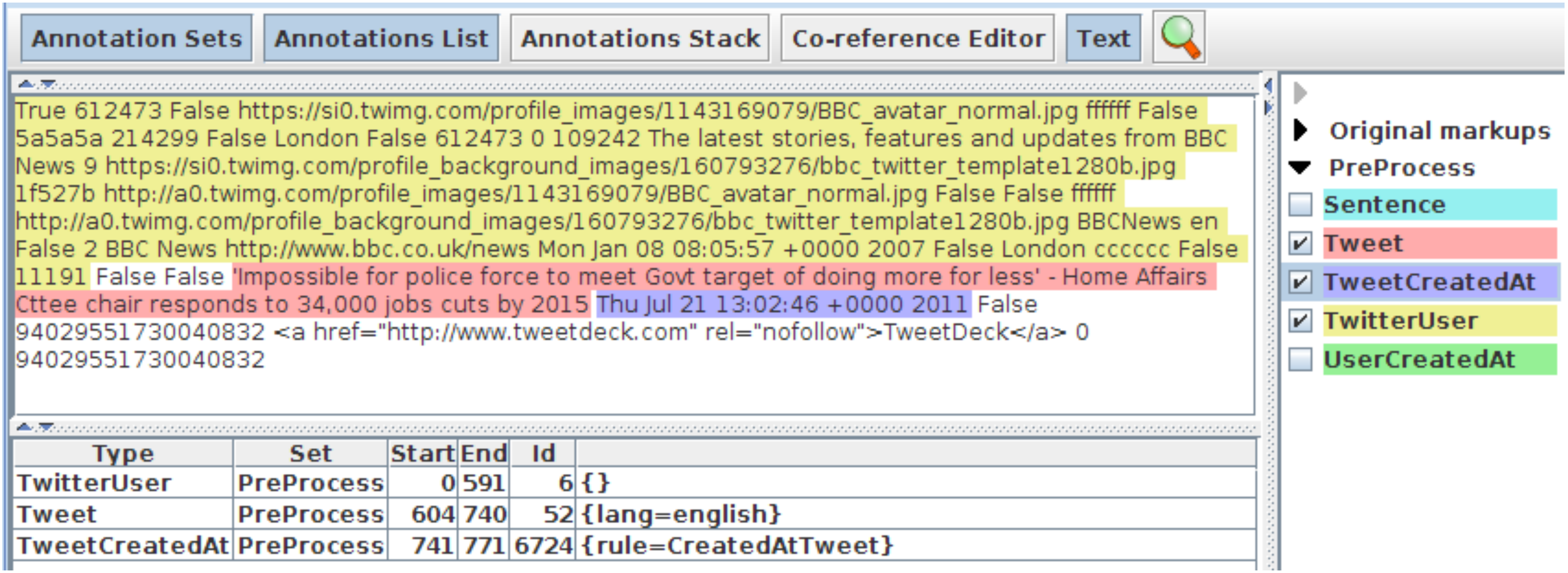
Without training, the langid.py tool does better than other language ID systems on social media

- Consistent improvement over plain TextCat, LangDetect and CLD
- Limited to no training data available for the 97 target languages

Hands-On 1: Language ID

- Load **twitie-lang-id.xgapp** in GATE (Restore Application From File)
- Create a new corpus, save to a serial datastore
- Load **lang-id-test-tweets.xml:**
- Choose **Populate from single file**, set root element to **doc_root**
- Run the application
- The Annotation Set Transfer first copies the text annotation from the "Original markups" set as a Tweet annotation in the PreProcess annotation set
- The Tweet Language Identification PR adds a "lang" feature to the Tweet annotation in the PreProcess set
- Inspect the results
- Keep the app open for later, but close the corpus

Language ID Results: English Example



- Various annotations created by the metadata-based pre-processing jape (tweet-metadata-parser.jape in resources)
- Sentence is an annotation created to span the entire tweet text
- TwitterUser spans the entire user information in the tweet
- TweetCreatedAt – the timestamp of this tweet

Tokenisation: example

General accuracy on microblogs: 80%
Goal is to convert byte stream to readily-digestible word chunks.
Word bound discovery is a *critical* language processing task

Newswire: **The LIBYAN AID Team successfully shipped these broadcasting equipment to Misrata last August 2011, to establish an FM Radio station ranging 600km, broadcasting to the west side of Libya to help overthrow Gaddafi's regime.**

Twitter: **RT @JosetteSheeran: @WFP #Libya breakthru! We move urgently needed #food (wheat, flour) by truck convoy into western Libya for 1st time :D**

@ojmason @encoffeedrinker But it was #nowthatcherisdead that was confusing (and not just to non-UK people!)

RT @Huddy85 : @Mz_Twilightxxx *kisses your asssneezes after* Lol**

Ima get you will.i.am NOTHING IS GONNA STAND IN MY WAY =)

Tokenisation: issues

Social media text is generally not curated, and typographical errors are common

Improper grammar, e.g. apostrophe usage:

doesn't → does n't
doesnt → doesnt

- Introduces previously-unseen tokens

Smileys and emoticons

I <3 you → I & lt ; you
This piece ;,,(so emotional → This piece ; , , (so emotional

- Loss of information (sentiment)

Punctuation for emphasis

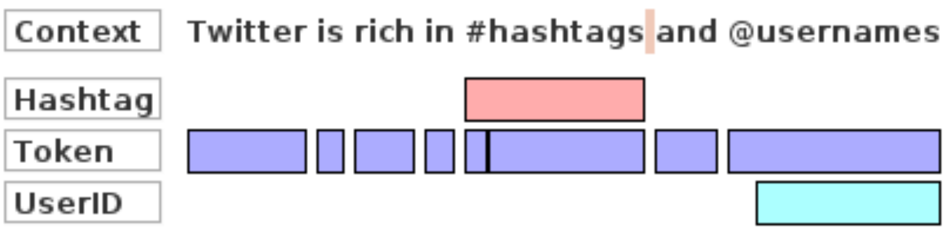
*HUGS YOU**KISSES YOU* → * HUGS YOU**KISSES YOU *

Words run together / skip

I wonde rif Tsubasa is okay..

Tokenisation: solutions

- We extend the Penn Treebank tool with twitter adaptations
- Layer multiple annotations on top of each other: Hashtags, Usernames



- Normalisation maps frequent nonstandard spellings to standard
- Via lookup dictionary (e.g. Han 2011); e.g. gonna → going to
- Regular expressions for known smileys/emoticons to avoid splitting them

- Segmenting individual hashtags is possible (Maynard 2014)

#openaccess → # open access

#palmoil → # palm oil

Hashtag analysis can be tricky

Even for humans!

- #nowthatcherisdead
- #powergenitalia
- #lesbocages
- #molestationnursery
- #teacherstalking
- #therapist

Test your social media skills!

What do these hashtags mean?

- #kktny
- #fomo
- #jomo
- #ootd
- #wcw

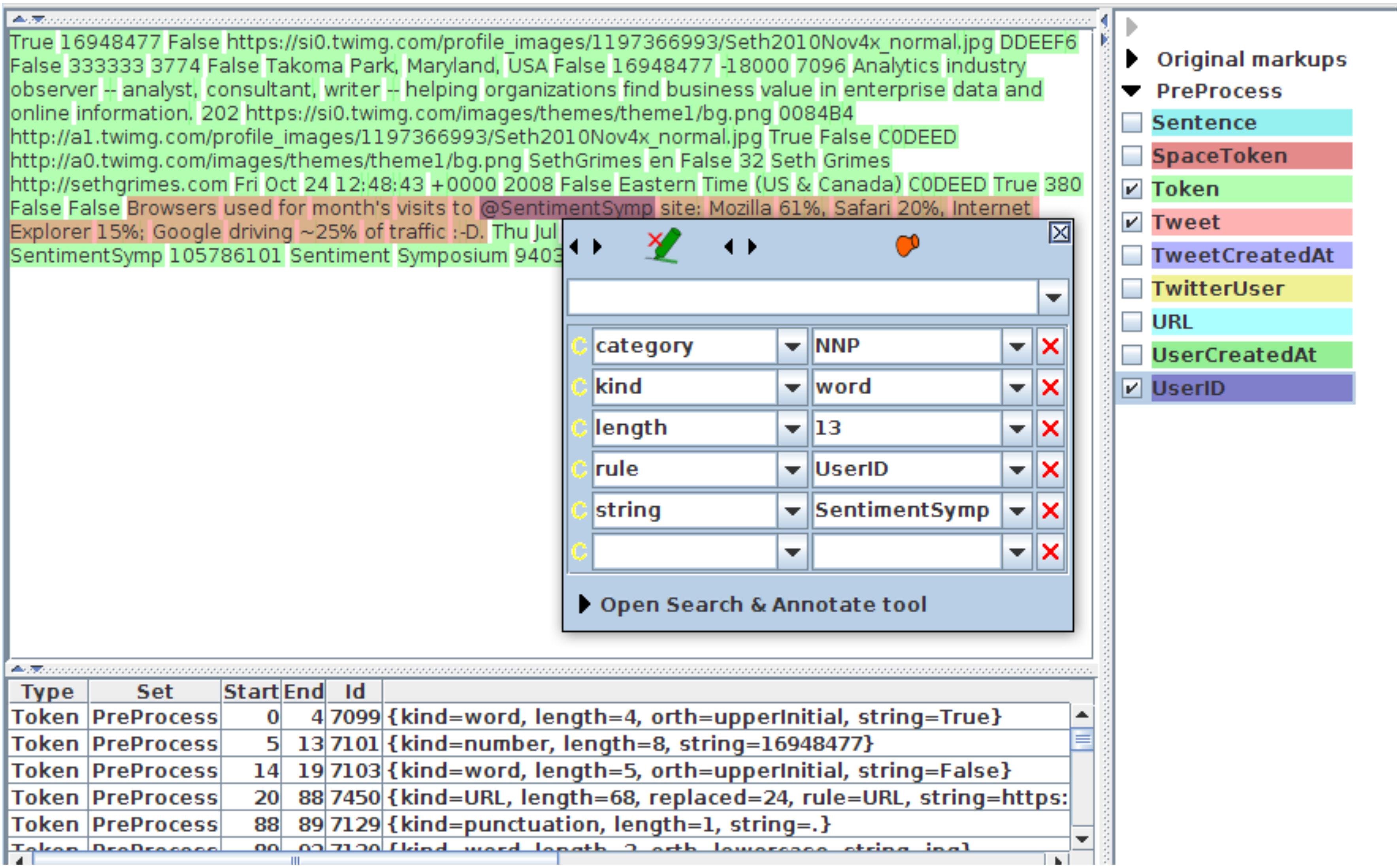
Hands-On: Hashtag and @mention tokenisation

- Load a Document Reset and Unicode Tokeniser
- Create a new application and add these to it (Reset first)
- Create a new corpus, name it “Tweets”
- Right-click on the corpus and select “populate from Twitter JSON”, selecting the file energy-tweets.json
- Look at the Token annotations in the Default annotation set
- Create a JAPE transducer, loading **resources/hashtag.jape**
- Add it to the application and re-run. Hashtag annotations appear
- Now add a new rule to detect @mentions as UserID annotations
- Right-click on the JAPE transducer, re-load, and re-run the app

The GATE Twitter Tokeniser

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

GATE Twitter Tokeniser: An Example

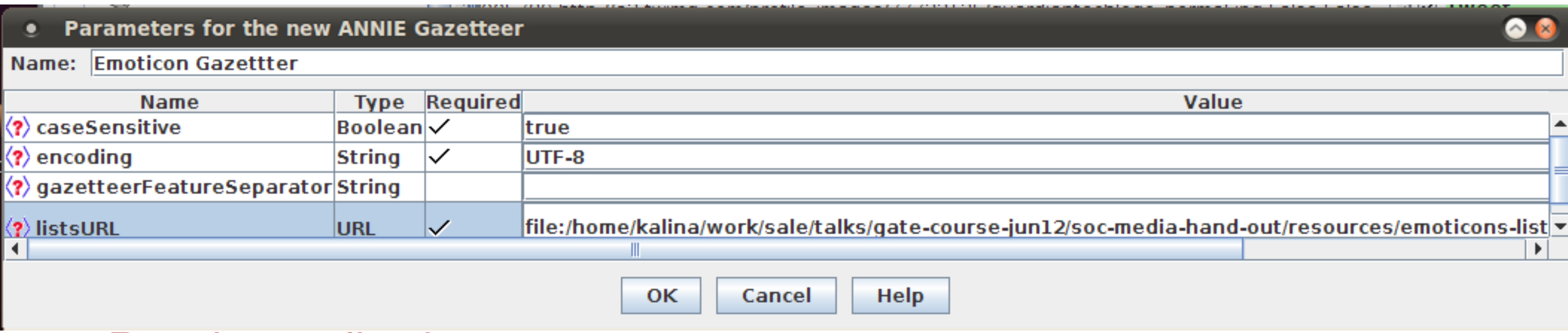


Hands-on: Running GATE's Tweet Tokeniser

- Create a new application, call it Twitter App
- Load a Document Reset and Twitter Tokeniser
- Run app on your energy tweets and inspect results (Hashtag, UserID)
- This should give you roughly the same results
- Take a quick look at the actual rules for Hashtag and UserID recognition in tokeniser/twitter.jape. See how they differ from the simple ones we wrote earlier.

Emoticon Detection

- There is a gazetteer list of some commonly used emoticons in your hand-outs, resources/emoticons-list.
- Create an ANNIE Gazetteer PR, name it Emoticon gazetteer
- Change the default separator from : to \t (colons are often in smileys)
- Set the listsURL to the emoticons-lists.def file



- Run the application
- Inspect the Lookup annotations in GATE Developer

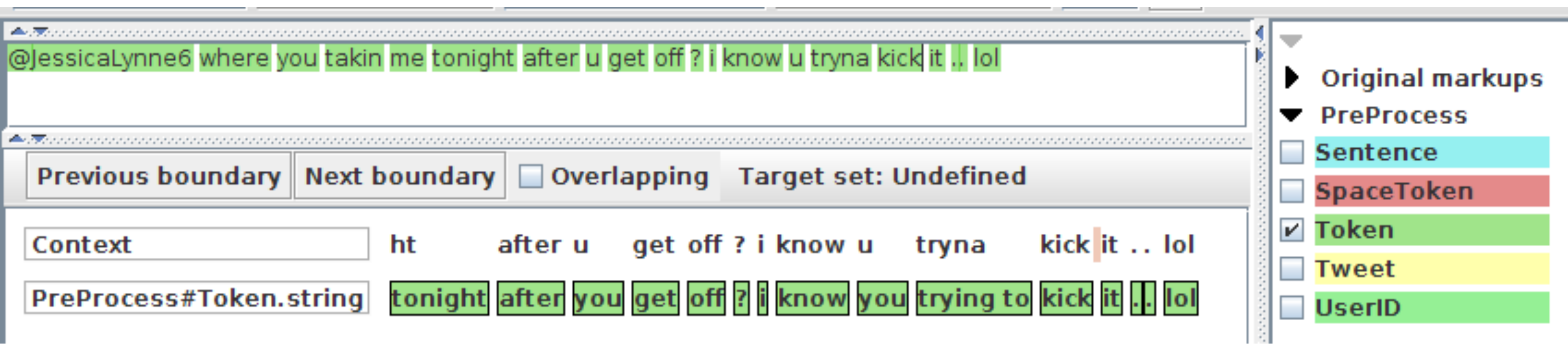
Tweet Normalisation

- “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 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

Lexical normalisation

- Two classes of word not in dictionary
  - 1. Mis-spelled dictionary words
  - 2. Correctly-spelled, unseen words (e.g. foreign surnames)
- Problem: Mis-spelled unseen words (these can be in the dict!)
- 1st challenge: separate out-of-vocabulary and in-vocabulary
- 2nd challenge: fix mis-spelled IV words
- Edit distance (e.g. Levenshtein): count character adds, removes
  - zseged → szeged (distance = 2)
  - Pronunciation distance (e.g. double metaphone):
  - YEEAAAHHH → yeah
- Need to set bounds on these, to avoid over-normalising OOV words

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

GATE Tweet Normaliser

- Load the Tweet Normaliser PR
- Add it at the end of your pipeline
- Run the pipeline and inspect the results
- Check the features on **Token** annotations
- If you can't find any normalised words, just edit one of the tweets and add your own slang words to normalise!

Part-of-speech tagging: example

- Many unknowns:

Music bands: Soulja Boy | TheDeAndreWay.com in stores Nov 2, 2010

Places: #LB #news: Silverado Park Pool Swim Lessons

- Capitalisation issues:

@thewantedmusic on my tv :) aka derek

last day of sorting pope visit to birmingham stuff out

Don't Have Time To Stop In??? Then, Check Out Our Quick Full Service Drive Thru Window :)

Part-of-speech tagging: example

- Slang
- **~HAPPY B-DAY TAYLOR !!! LUVZ YA**

- Orthographic errors
- **dont even have homwork today, suprising?**

- Dialect

fancy a cheeky nandoz tho

Can I have a go on your iPad?

Part-of-speech tagging: issues

Unknown words fall roughly into two categories

- Standard token, non-standard orthography;

freinds

KHAAAANNNNNN!

- Non-standard token, standard orthography

omg + bieber → omb

Huntingdon / Huntington

Load & configure the Stanford Tagger

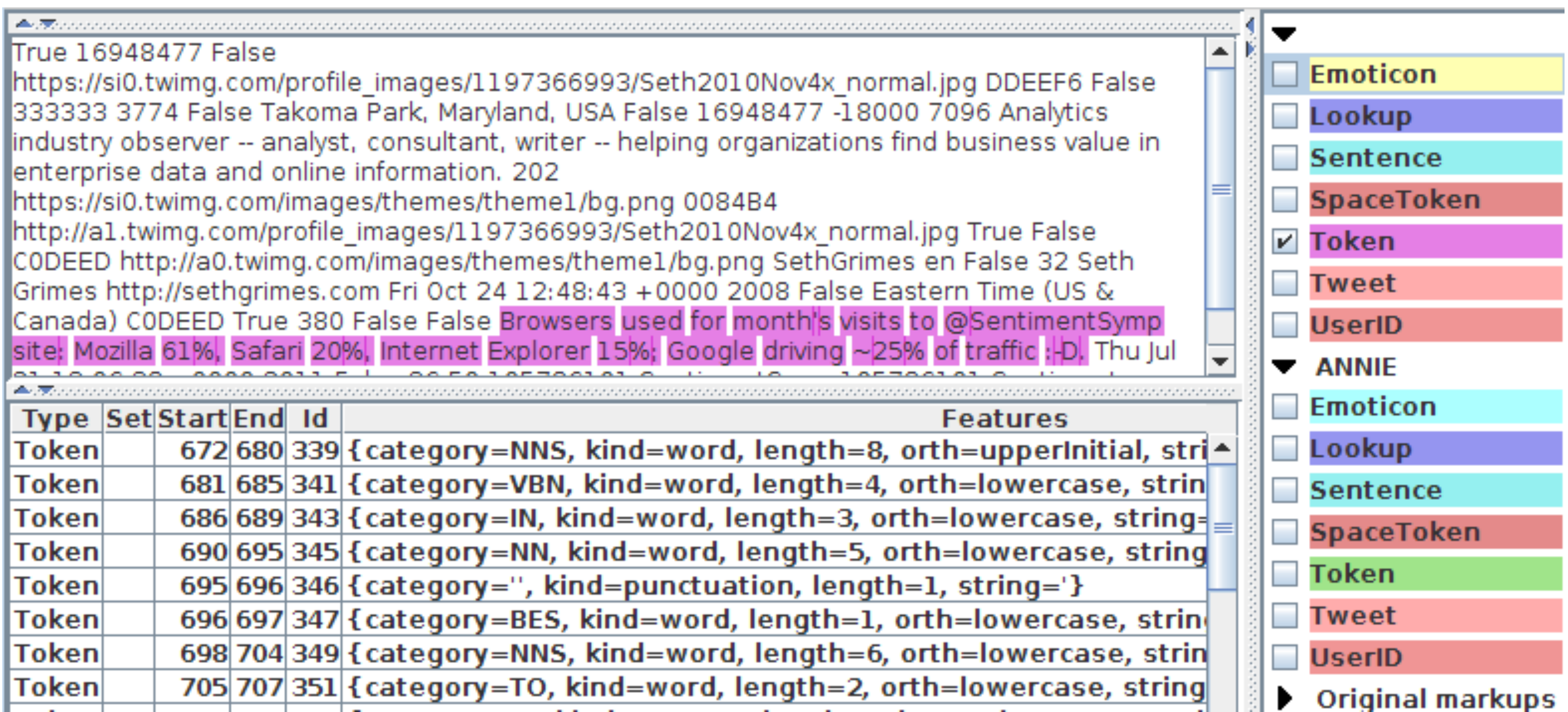
- Load the StanfordCoreNLP plugin through the Plugin Manager
- Create an instance of Stanford POS Tagger with this model: **resources/gate-EN-twitter.model**
- Add to the end of the application and run it

Let’s compare ANNIE and TwitIE

- Load the ANNIE application
- Change the annotationSetName, inputAS and outputAS parameters to ANNIE for every PR
- Run it
- Now, carefully, go to your TwitIE application and set the Document Reset parameters to keep the ANNIE annotation set (setsToKeep – add ANNE to the list)
- Otherwise, it would get removed when we run TwitIE
- Now run TwitIE again

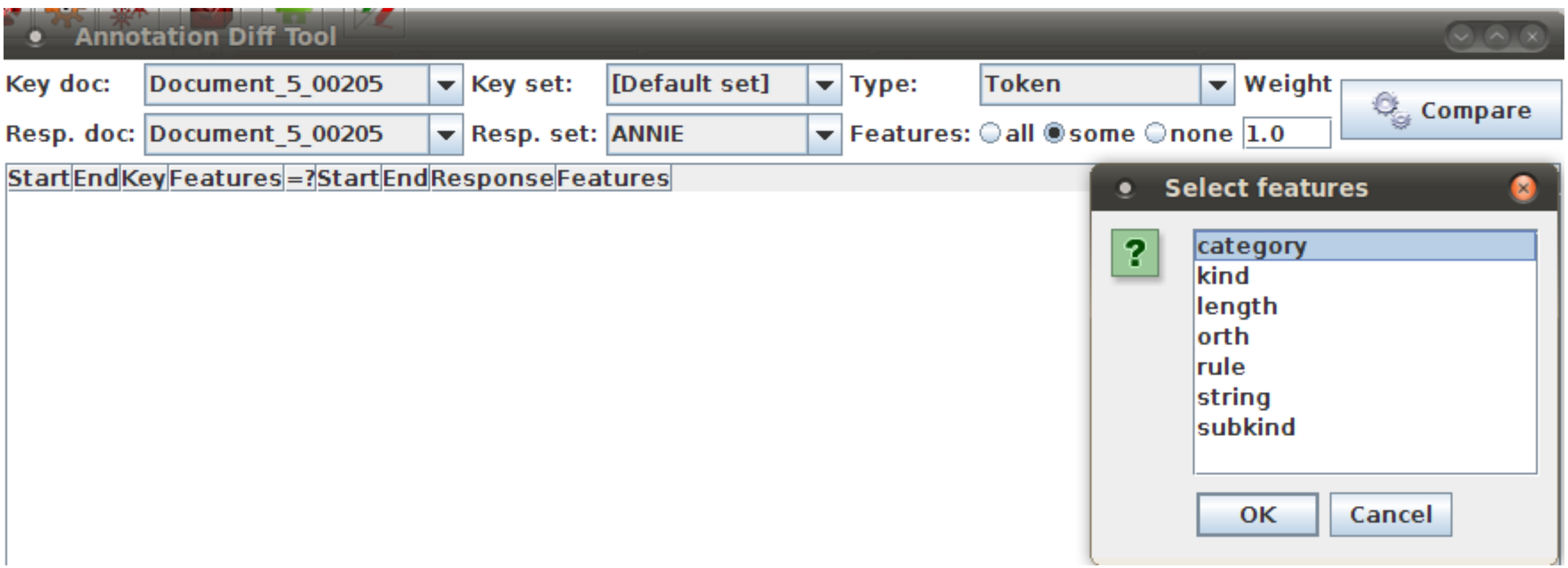
TwitIE POS Tagger Results: Example

- You should get results in 2 sets:
  - ANNIE will have the POS tags from the ANNIE POS Tagger
  - The default set will have those from the TwitIE Tagger



Compare Differences: Annotation Diff

- Click on the Annotation Diff button 
- Select a document from the test corpus (same Key and Resp)
- Key set: [Default set]; Resp. set: ANNIE
- Type: Token; Features: some, then select: category



Compare Differences (2)

- Click on the Compare button
- Inspect the results; repeat for 1-2 more documents
- HINT: Clicking on the Start column will sort tokens by offset

Start	End	Key	Features	=?	Start	End	Response	Features
736	741	Nokia	{category=NNP, kind=...gth=5, string=Nokia}	=	736	741	Nokia	{category=NNP, kind=...gth=5, string=Nokia}
742	747	Posts	{category=VBZ, kind=...gth=5, string=Posts}	=	742	747	Posts	{category=VBZ, kind=...gth=5, string=Posts}
748	752	Huge	{category=JJ, kind=w...ngth=4, string=Huge}	=	748	752	Huge	{category=JJ, kind=w...ngth=4, string=Huge}
753	762	Quarterly	{category=JJ, kind=w...9, string=Quarterly}	=	753	762	Quarterly	{category=JJ, kind=w...9, string=Quarterly}
763	767	Loss	{category=NN, kind=w...ngth=4, string=Loss}	=	763	767	Loss	{category=NN, kind=w...ngth=4, string=Loss}
767	768	,	{string=,, length=1,...tuation, category=,}	=	767	768	,	{string=,, length=1,...tuation, category=,}
769	773	Sees	{category=VBZ, kind=...ngth=4, string=Sees}	<>	769	773	Sees	{category=NNP, kind=...ngth=4, string=Sees}
774	780	Better	{category=NNP, kind=...th=6, string=Better}	=	774	780	Better	{category=NNP, kind=...th=6, string=Better}
781	786	Times	{category=NNP, kind=...gth=5, string=Times}	=	781	786	Times	{category=NNP, kind=...gth=5, string=Times}
787	792	Ahead	{category=NNP, kind=...gth=5, string=Ahead}	=	787	792	Ahead	{category=NNP, kind=...gth=5, string=Ahead}
793	794	-	{category=:, subkind... length=1, string=-}	=	793	794	-	{category=:, subkind... length=1, string=-}
795	819	http://on...	{rule=URL, temp_cate...ed=13, category=URL}	<>	795	819	http://o...	{rule=URL, temp_cate...ced=13, category=CD}

- We are still improving the tweet POS model, but major improvements make it current state-of-the-art