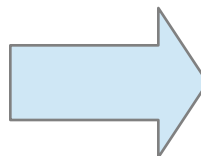


GATE and Social Media: Language ID, tokenisation and hashtags

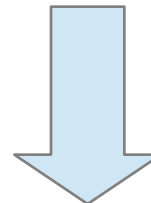
Dominic Rout
Kalina Bontcheva
Leon Derczynski (slides)



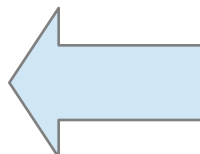
Text



Language ID



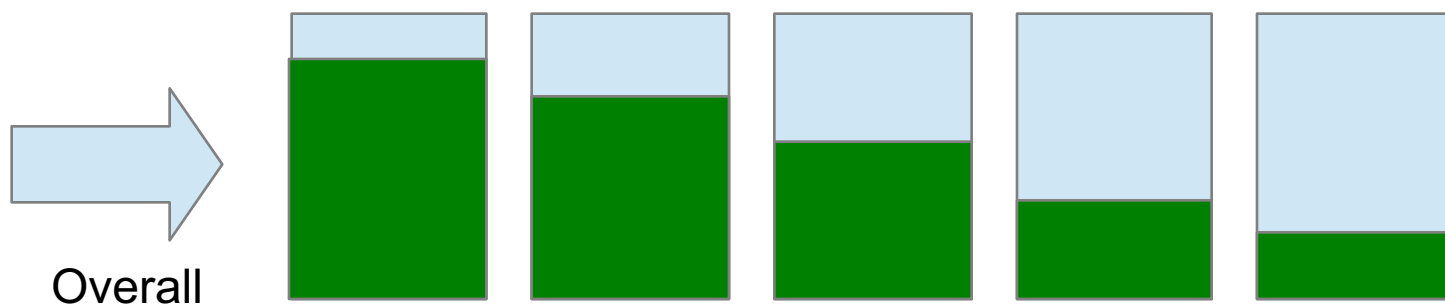
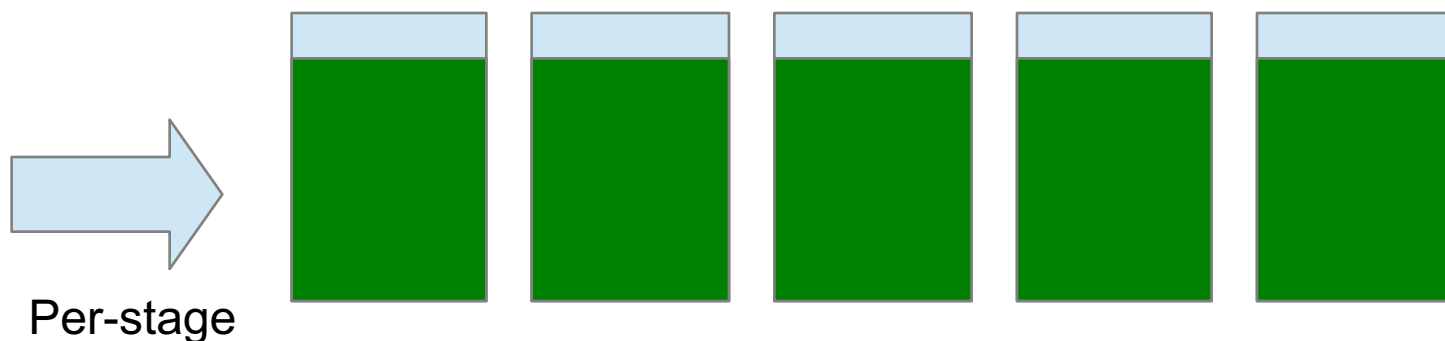
Tokenisation



**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 intended to be.

Newsire:

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 intended to be.

Newsire: 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

General accuracy on microblogs: 89.5% (Preotiuc-Pietro 2012)

Compared to accuracy on formal text: 99.4% (Carter 2013)

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

- 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 sources of 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.

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

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

The variety of data and sparsity of features makes selection important

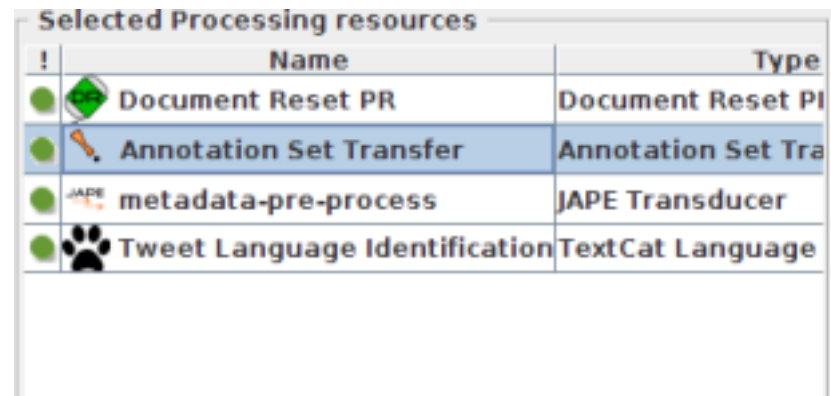
- LD focuses on task-relevant features using information gain
- Features with a high LD score are informative about language, without being informative about domain
- Candidate features pruned before applying LD based on term frequency





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

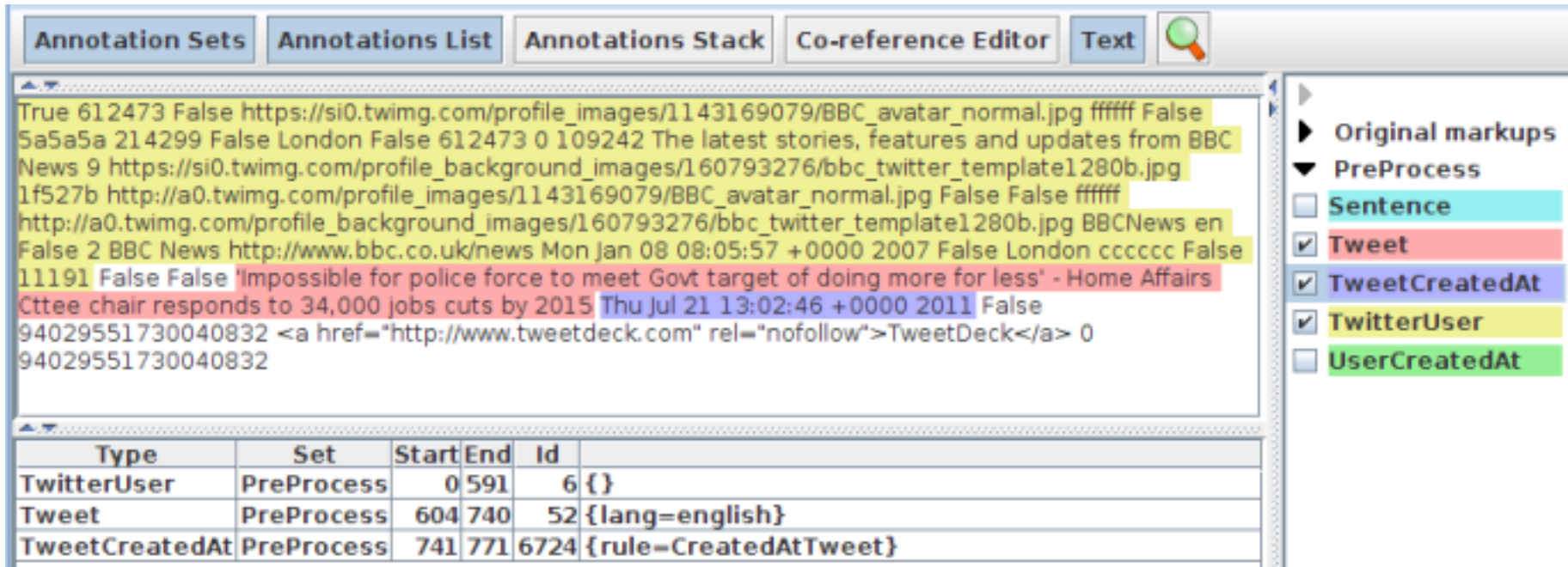
Hands-On 1: Language ID

- Load **twitie-lang-id.xgapp** in GATE (Restore Application From File)
- Create a new corpus, save to DS and load **lang-id-small-test-set.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
- Close the corpus

A screenshot of the 'Selected Processing resources' window in the GATE software. It displays a table with four rows of processing resources. The first row is 'Document Reset PR' (Document Reset PR). The second row is 'Annotation Set Transfer' (Annotation Set Transfer). The third row is 'metadata-pre-process' (JAPE Transducer). The fourth row is 'Tweet Language Identification' (TextCat Language).

!	Name	Type
	Document Reset PR	Document Reset PR
	Annotation Set Transfer	Annotation Set Transfer
	metadata-pre-process	JAPE Transducer
	Tweet Language Identification	TextCat Language

Language ID Results: English Example



The screenshot shows the GATE software interface with the 'Text' tab selected. The main text area displays a tweet with various annotations. The tweet text is: "True 612473 False https://si0.twimg.com/profile_images/1143169079/BBC_avatar_normal.jpg ffffff False 5a5a5a 214299 False London False 612473 0 109242 The latest stories, features and updates from BBC News 9 https://si0.twimg.com/profile_background_images/160793276/bbc_twitter_template1280b.jpg 1f527b http://a0.twimg.com/profile_images/1143169079/BBC_avatar_normal.jpg False False ffffff http://a0.twimg.com/profile_background_images/160793276/bbc_twitter_template1280b.jpg BBCNews en False 2 BBC News http://www.bbc.co.uk/news Mon Jan 08 08:05:57 +0000 2007 False London cccccc False 11191 False False 'Impossible for police force to meet Govt target of doing more for less' - Home Affairs Cttee chair responds to 34,000 jobs cuts by 2015 Thu Jul 21 13:02:46 +0000 2011 False 94029551730040832 TweetDeck 0 94029551730040832".

Below the text area, there is a table showing the annotations:

Type	Set	Start	End	Id	
TwitterUser	PreProcess	0	591	6	{ }
Tweet	PreProcess	604	740	52	{lang=english}
TweetCreatedAt	PreProcess	741	771	6724	{rule=CreatedAtTweet}

On the right side, there is a panel titled 'Original markups' and 'PreProcess'. The 'PreProcess' section has several checkboxes: 'Sentence' (unchecked), 'Tweet' (checked), 'TweetCreatedAt' (checked), 'TwitterUser' (checked), and 'UserCreatedAt' (unchecked).

- 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

Newsire:

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 ass**sneezes 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..`

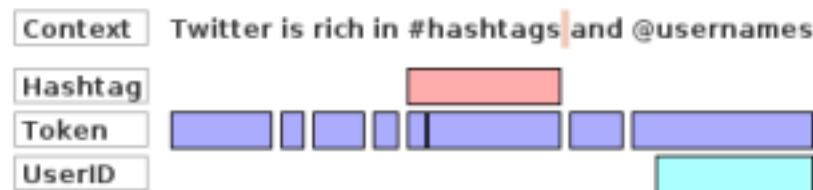
Tokenisation: solutions

O'Connor et al. (2010) apply a regular expression tokeniser to tweets, with the following adaptations:

- Treat #hashtags, @mentions, abbreviations, strings of punctuation, emoticons and UTF glyphs as single tokens
- Made available as “ttokenizer” tool

Bontcheva et al. (2013) 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](#)
 - [#swankkkkk](#) → <#> [swan](#) [kkk](#) [k](#) [k](#) [?](#)

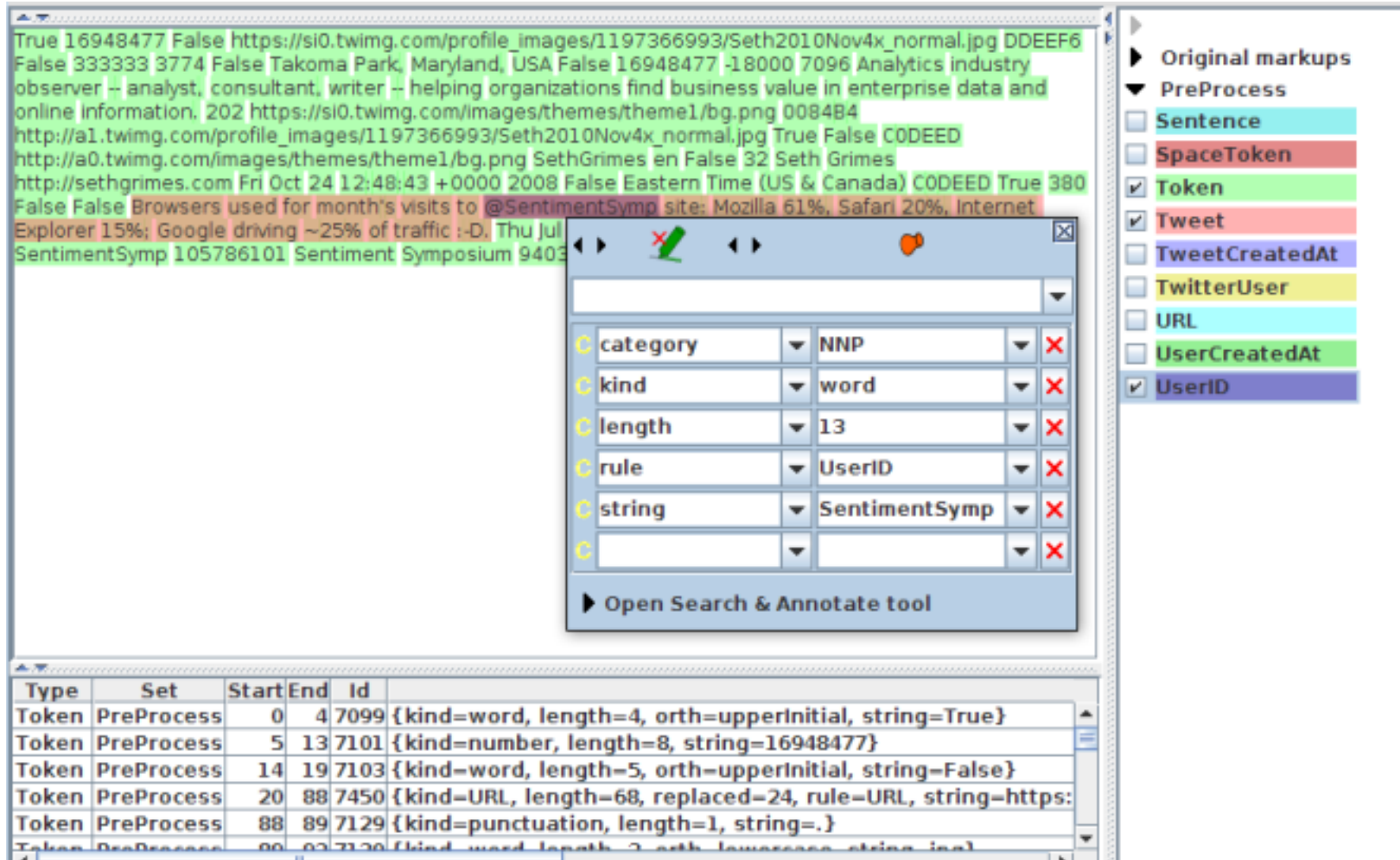
Hands-On: Hashtag and @mention tokenisation

- Load the **GATE Unicode Tokeniser** PR, with its default settings
- Load a **Document Reset** PR with defaults
- Create a new corpus pipeline app; add Reset, then the Tokeniser
- Create a new corpus and populate from single concatenated file, using **test-10-tweets.xml** (Root element: **doc_root**)
- Inspect the results, especially around hashtags and @mentions
- It helps to show the “text” annotation from the “Key” AS
- 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



The screenshot displays the GATE Twitter Tokeniser interface. The main window shows a tweet being processed, with various fields and buttons. A search and annotation tool is open, showing a table of tokens. The right sidebar lists the processing steps and the current state of the tweet.

Original markups

- PreProcess
- Sentence
- SpaceToken
- ☒ Token
- ☒ Tweet
- TweetCreatedAt
- TwitterUser
- URL
- UserCreatedAt
- ☒ UserID

Search & Annotate tool

category	NNP	kind	word	length	13	rule	UserID	string	SentimentSymp

Open Search & Annotate tool

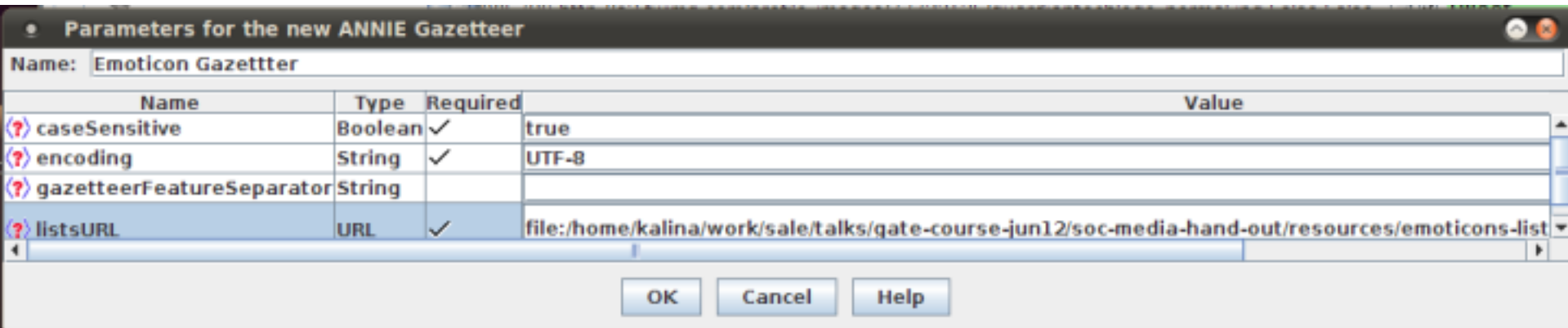
Type	Set	Start	End	Id	Details
Token	PreProcess	0	4	7099	{kind=word, length=4, orth=upperInitial, string=True}
Token	PreProcess	5	13	7101	{kind=number, length=8, string=16948477}
Token	PreProcess	14	19	7103	{kind=word, length=5, orth=upperInitial, string=False}
Token	PreProcess	20	88	7450	{kind=URL, length=68, replaced=24, rule=URL, string=https:}
Token	PreProcess	88	89	7129	{kind=punctuation, length=1, string=.
Token	PreProcess	90	92	7130	{kind=word, length=2, orth=lowercase, string=inc}

Hands-on: Running GATE's Tweet Tokeniser

- Right click on Processing Resources, load ANNIE English Tokeniser
 - Leave TokeniserRulesURL unchanged
 - For **TransducerGrammarURL** navigate to your hands-out directory, then choose **resources/tokeniser/twitter.jape**
- Add this Tweet Tokeniser at the end of the **TwitIE tutorial app**
- Set the AnnotationSetName parameter to **PreProcess**
- Run app on the 10 tweets and inspect results (Hashtag, UserID)
- Note that the Token annotations under UserIDs have now PoS category NNP, since they are proper names
- Take a quick look at the actual rules for Hashtag and UserID recognition in 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
- Remove the default separator (colon : is often in smileys)
- Point it to the emoticons directory



Name	Type	Required	Value
caseSensitive	Boolean	✓	true
encoding	String	✓	UTF-8
gazetteerFeatureSeparator	String		
listsURL	URL	✓	file:/home/kalina/work/sale/talks/gate-course-jun12/soc-media-hand-out/resources/emoticons-list

- Add it at the end of your pipeline, set the AnnotationSetName parameter to PreProcess, run the app
- Inspect the Lookup annotations in GATE Developer