

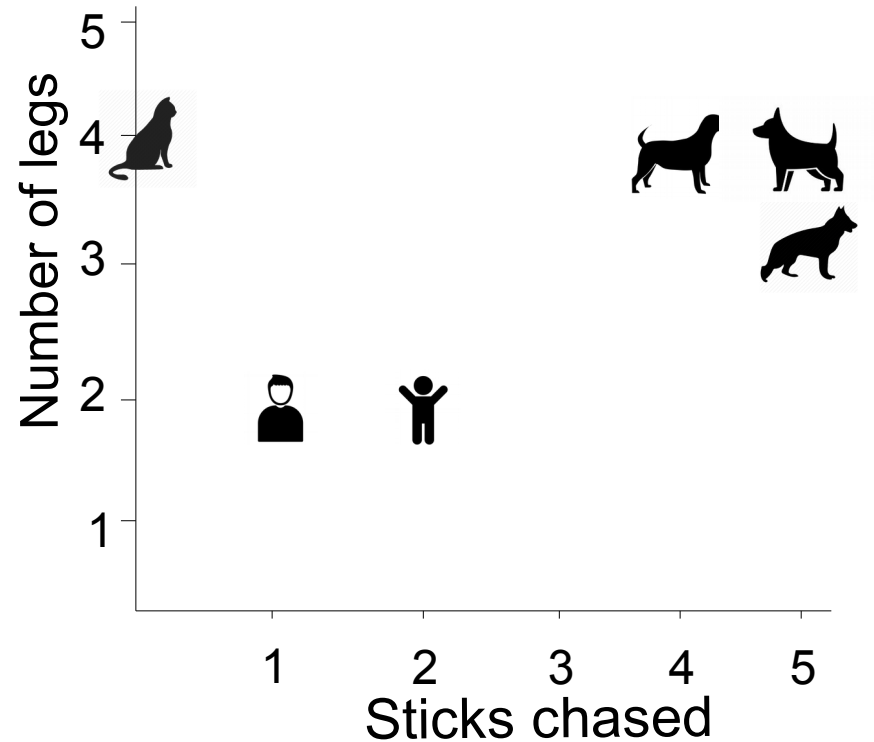
A bit of theory: Algorithms

- There are different kinds of algorithms
 - Vector space models. e.g. support vector machines
 - Decision trees, e.g. C45
 - Probabilistic models, e.g. Naive Bayes
 - Neural networks, e.g. CNN, RNN (more about these later)
- Time is way too short to go into them, but Andrew Ng's course is popular:
`https://www.coursera.org/learn/machine-learning`
- Let's talk briefly about vector space models

Vector space models—Is it a dog?

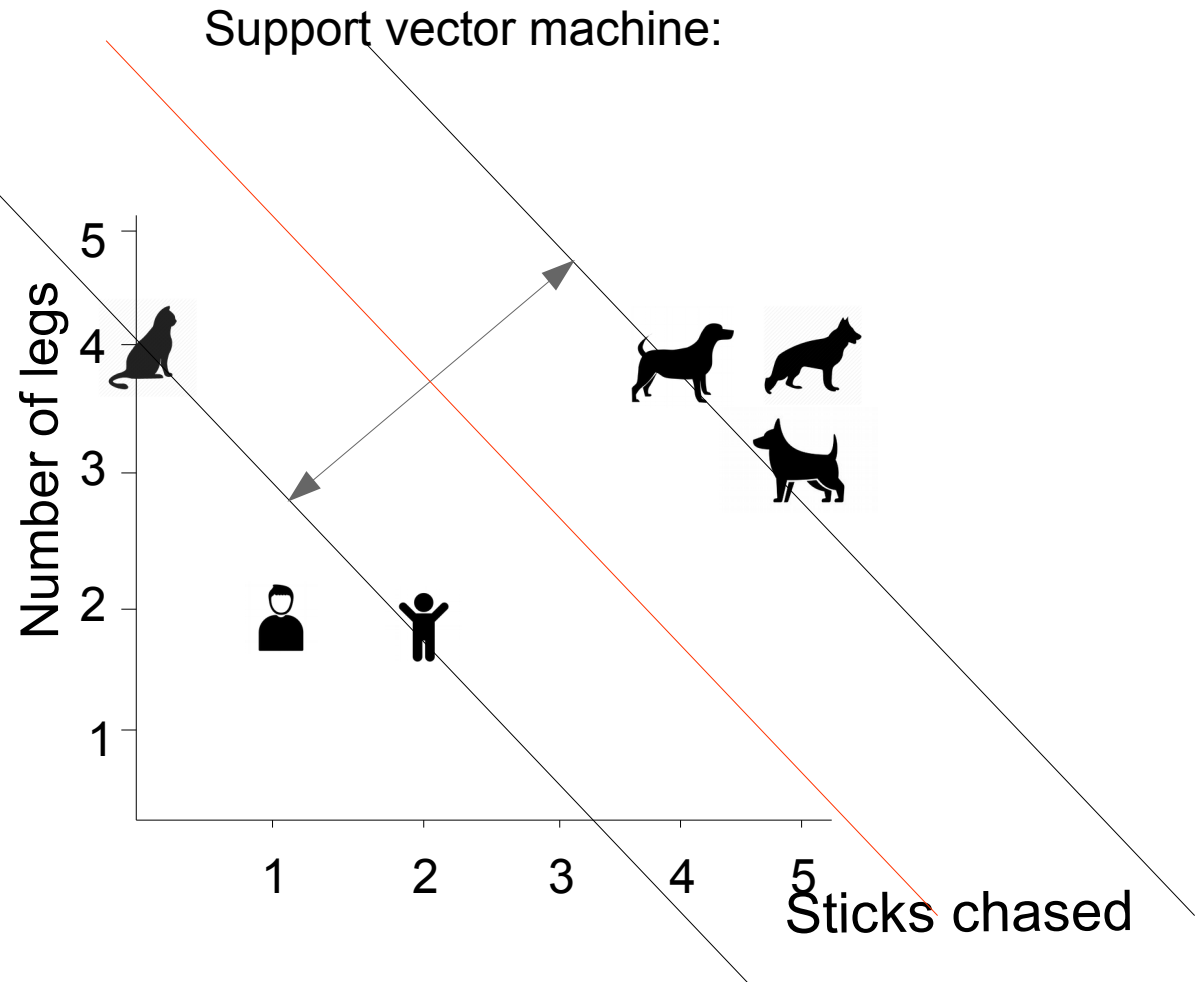
features	class
4,0	no
2,1	no
2,2	no
4,4	yes
4,5	yes
3,5	yes

Vector space model:



Support vector machine—Is it a dog?

- Find the biggest gap possible between dog and not-dog
- Kernel trick maps to higher dimensionality, makes rings and other separator shapes possible
- Not always possible/desirable to find perfect solution, hence cost parameter





Nominal features

- This dataset has two numeric features; number of legs and number of sticks chased
- What about words? How do they become numbers?
 - Assign an index to each unique word
 - Each word becomes a “one hot” vector, e.g. “love” becomes [0, 1, 0, 0, 0, 0, 0, 0]
 - Add them all together

features	class
4,0,no	
2,1,no	
2,2,no	
4,4,yes	
4,5,yes	
3,5,yes	

word indices	unique words
0	i
1	love
2	the
3	acme
4	500
5	sucks
6	is
7	best

features (all the words in the review) class

“I love the Acme 500”, positive
 “The Acme 500 sucks”, negative
 “The Acme 500 is the best”, positive

1,1,1,1,1,0,0,0,positive
 0,0,1,1,1,1,0,0,negative
 0,0,2,1,1,0,1,1,positive

Data formats: Sparse vs Dense

DENSE:

```
1,1,1,1,1,0,0,0,positive  
0,0,1,1,1,1,0,0,negative  
0,0,2,1,1,0,1,1,positive
```

SPARSE:

```
0:1,1:1,2:1,3:1,4:1,positive  
2:1,3:1,4:1,5:1,negative  
2:2,3:1,4:1,6:1,7:1,positive
```

- What happens on a really big corpus?
 - There are many possible words, and most of them don't occur, so we get huge numbers of zeros!
 - Can't we just specify the ones that aren't zeros? Yes; it's called “sparse” representation
- But do we really need a separate dimension for each word?
 - Not really. It might even help if similar words point in similar directions—word embeddings
 - So then we're back to dense vectors anyway
- In either case, we lost word order
 - Could use n-grams—short word sequences
 - Features become rarer though so can be harder to learn unless corpus is v. large

Chunking—Practical Exercise

Chunking for NER

- Chunking, as we saw at the beginning, means finding parts of text
- This task is often called Named Entity Recognition (NER), in the context of finding person and organization names
- The same principle can be applied to any task that involves finding where things are located in text
 - For example, finding the noun phrases
 - Can you think of any others?

California Governor Arnold Schwarzenegger proposes deep cuts.

Person

Chunking for NER

- It's implemented as a twist on classification (everything is classification under the hood!)
- We achieve this in the Learning Framework by identifying which tokens are the beginning of a mention, which are the insides and which are the outsides (“BIO”)
 - There are other schemes; the old Batch Learning PR used BE (beginnings and ends)
- You don't need to worry about the Bs, Is and Os; the Learning Framework will take care of all that for you! You just need a corpus annotated with entities

California Governor Arnold Schwarzenegger proposes deep cuts.





Chunking—Practical Exercise

- Materials for this exercise are in the folder called “chunking-hands-on”
- You might want to start by closing any applications and corpora from the previous exercise, so we have a fresh start

Finding Person Mentions using Chunking Training and Application PRs



Load the corpus

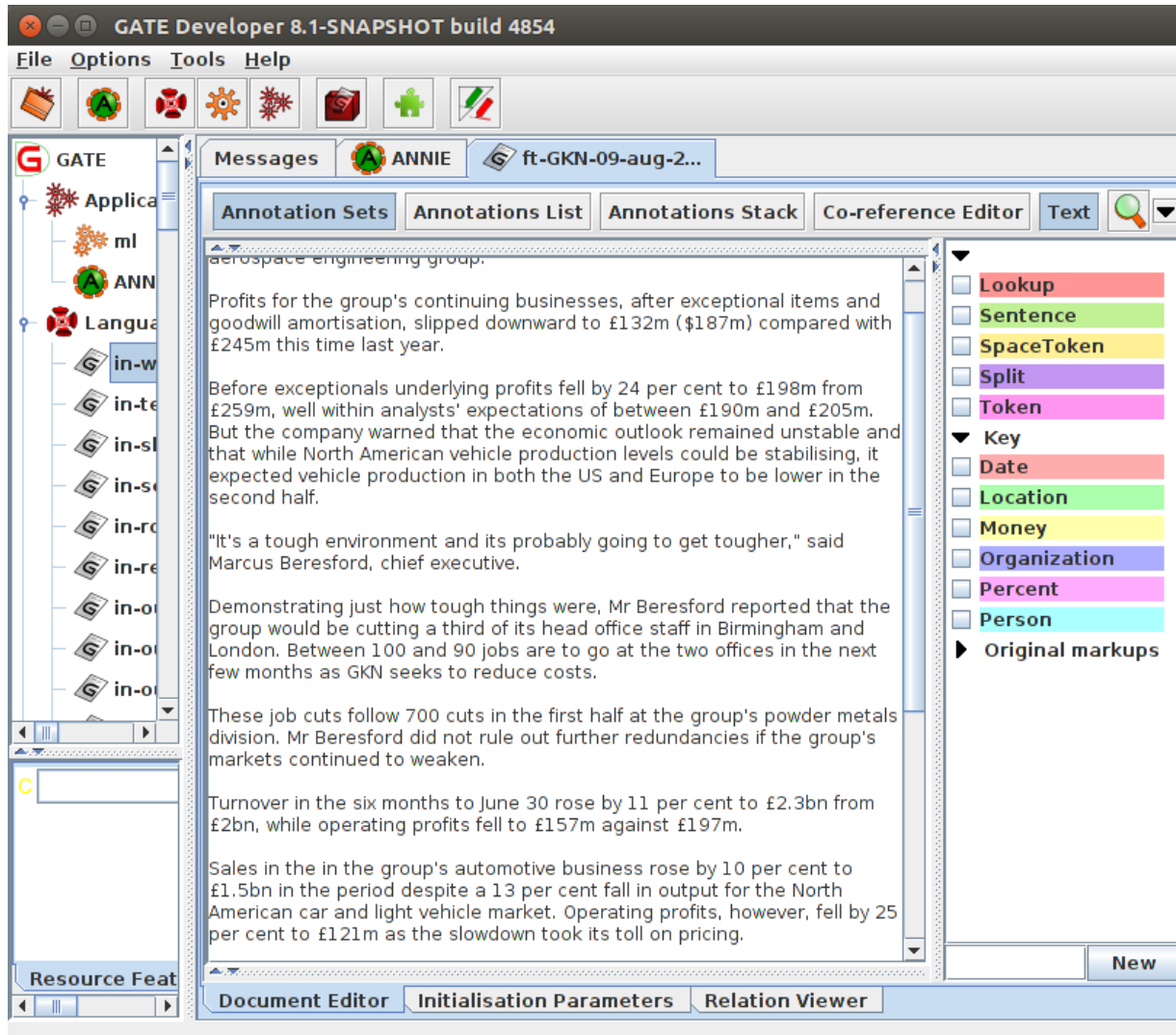
- Create corpora for training and testing, with sensible names
- Populate them from the training and testing corpora you have in your chunking hands on materials
- Open a document and examine its annotations



Examining the corpus

- The corpus contains an annotation set called “Key”, which has been manually prepared
- Within this annotation set are annotations of types “Date”, “Location”, “Money”, “Organization” and so forth

Creating the application



- As previously, if we run ANNIE on the corpus, we have more annotations to work with
- So start by loading ANNIE as the basis for your application
- Again, we don't need the NE transducer or orthomatcher

NER GATE application



GATE Developer 8.2-SNAPSHOT build 5490

File Options Tools Help

GATE

- Applications
 - ANNE
 - Language Resources
 - Processing Resources
 - Annotation Set Transfer
 - LF_ApplyChunking 0001
 - LF_TrainChunking 0003
 - ANNE OrthoMatcher
 - ANNE NE Transducer
 - ANNE POS Tagger
 - ANNE Sentence Splitter
 - ANNE Gazetteer
 - ANNE English Tokeniser
 - Document Reset PR
 - Datatypes

Messages ANNE

Loaded Processing resources

Name	Type
ANNE NE Transducer	ANNE NE Transducer
ANNE OrthoMatcher	ANNE OrthoMatcher
LF_ApplyChunking 00031	LF_ApplyChunking

Selected Processing resources

Name	Type
Document Reset PR	Document Reset PR
ANNE English Tokeniser	ANNE English Tokeniser
ANNE Gazetteer	ANNE Gazetteer
ANNE Sentence Splitter	ANNE Sentence Splitter
ANNE POS Tagger	ANNE POS Tagger
Annotation Set Transfer 00036	Annotation Set Transfer
LF_TrainChunking 00030	LF_TrainChunking

Run "Annotation Set Transfer 00036"?

Yes No If value of feature is

Corpus: <none>

Runtime Parameters for the "Annotation Set Transfer 00036" Annotation Set Transfer:

Name	Type	Required	Value
annotationTypes	ArrayList		[]
copyAnnotations	Boolean	✓	false
inputASName	String		
outputASName	String		
tagASName	String		Original markups
textTagName	String		

Run this Application

Serial Application Editor Initialisation Parameters About...

Annotation Set Transfer 00036 loaded in 0.001 seconds

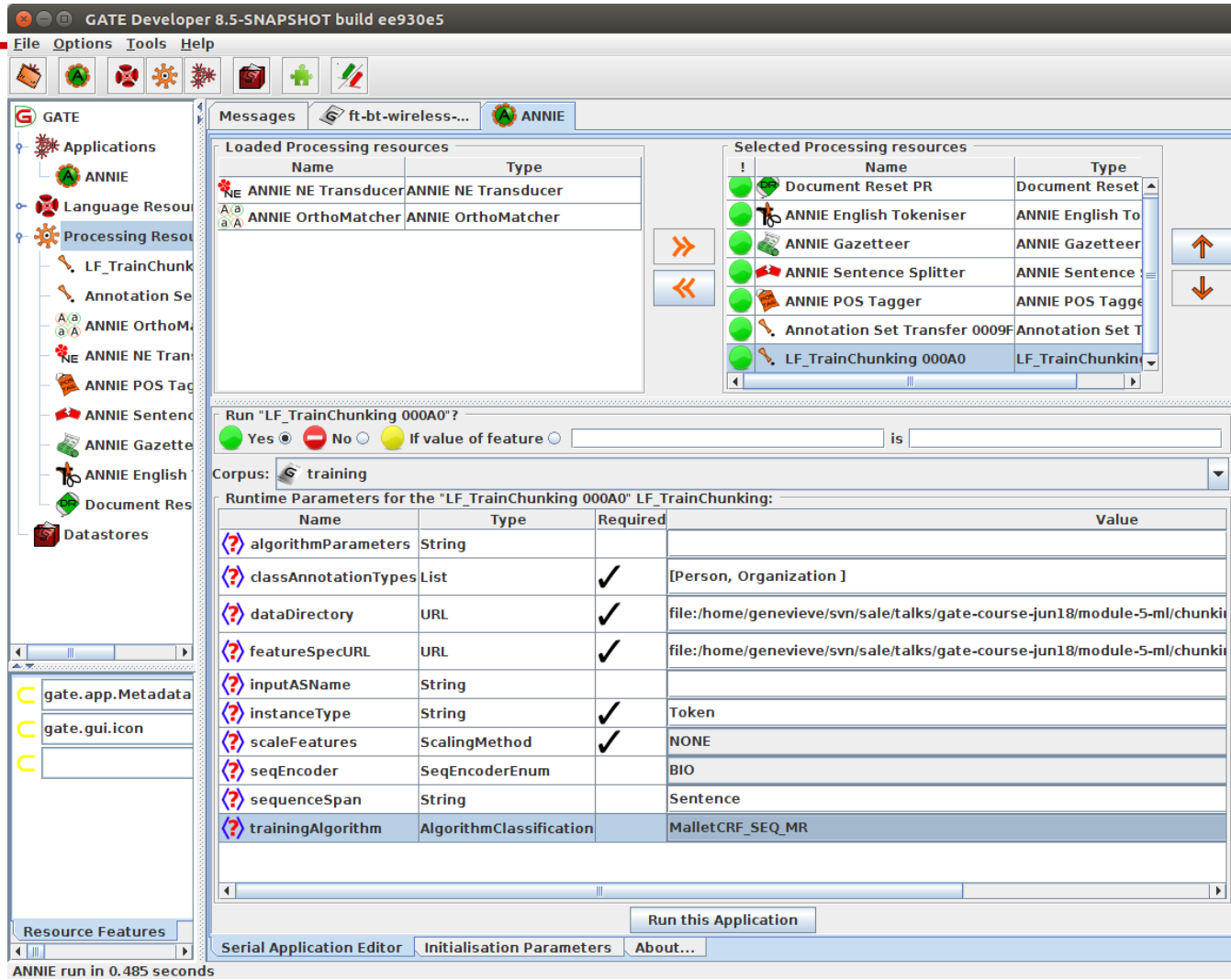
- Again, we need an Annotation Set Transfer, so create and add one
- Then create training chunking PR



Annotation Set Transfer

- We'll use the annotation set transfer to copy the Person and Organization annotations up to the default annotation set, where we can learn them
- **Go ahead and set up your AST now**
- Be sure to copy them, not move them!

Chunking training parameters



The screenshot shows the GATE Developer interface with the following components:

- Left Panel:** A tree view showing the project structure, including Applications, ANNIE, Language Resources, Processing Resources, and Datastores.
- Messages Panel:** Shows the selected processing resources for the ANNIE application.
- Selected Processing resources:** A list of resources including Document Reset PR, ANNIE English Tokeniser, ANNIE Gazetteer, ANNIE Sentence Splitter, ANNIE POS Tagger, Annotation Set Transfer 0009F, and LF_TrainChunking 000A0.
- Run Dialog:** A dialog box titled "Run 'LF_TrainChunking 000A0'?" with a "Yes" button selected and a "Corpus:" dropdown set to "training".
- Runtime Parameters Table:** A table showing the configuration for the "LF_TrainChunking 000A0" resource.

Name	Type	Required	Value
algorithmParameters	String		
classAnnotationTypes	List	✓	[Person, Organization]
dataDirectory	URL	✓	file:/home/genevieve/svn/sale/talks/gate-course-jun18/module-5-ml/chunki
featureSpecURL	URL	✓	file:/home/genevieve/svn/sale/talks/gate-course-jun18/module-5-ml/chunki
inputASName	String		
instanceType	String	✓	Token
scaleFeatures	ScalingMethod	✓	NONE
seqEncoder	SeqEncoderEnum		BIO
sequenceSpan	String		Sentence
trainingAlgorithm	AlgorithmClassification		MalletCRF_SEQ_MR
- Bottom Panel:** Shows the "Run this Application" button and tabs for "Serial Application Editor", "Initialisation Parameters", and "About...".

At the bottom of the window, it says "ANNIE run in 0.485 seconds".

- Let's look at the parameters for the training PR
- Instead of targetFeature, we have classAnnotationTypes

Chunking training parameters

- For **classification**, the class to learn is in a feature on the instance, is specified to the PR in the targetFeature parameter
- But for **chunking**, the class or classes to learn take the form of an annotation type.
 - The advantage to learning Person and Organization both at once is that if it's a Person it can't be an Organization
- This type to learn is indicated in the classAnnotationTypes parameter



Chunking training parameters

- Set the `classAnnotationTypes` now
- Set the `dataDirectory` to where you want to save your model, and set the `featureSpecURL` (there's a feature spec to get you started in the hands on materials)
- Set `instanceType`. What do you think it should be?

Sequence Spans

- sequenceSpan is only relevant when using sequence learners
- Sequence learners classify each instance in the span by making use of the others
- For example, a noun phrase might be more likely to follow a determiner than a preposition, or a person name might be more likely to follow the word “Mrs”
- **We'll try the Conditional Random Fields sequence learner**
 - You don't have to use a sequence learner for chunking though
- What do you think would be a good sequence span?

Sequence Spans

- Sequence spans should be spans within which instance classes follow patterns
 - For example, grammatical rules apply to sequences of parts of speech
 - However, sentiment classifications of individual customer reviews don't form a meaningful sequence
- A sequence span shouldn't be longer than necessary
- Sentence would be a good span for our task
- Fortunately, ANNIE creates sentence annotations for us, so those are available to use
- **Set `sequenceSpan` to "Sentence"**

Feature Specification

```
<ML-CONFIG>
```

```
<ATTRIBUTE>  
<TYPE>Token</TYPE>  
<FEATURE>category</FEATURE>  
<DATATYPE>nominal</DATATYPE>  
</ATTRIBUTE>
```

```
<ATTRIBUTE>  
<TYPE>Token</TYPE>  
<FEATURE>kind</FEATURE>  
<DATATYPE>nominal</DATATYPE>  
</ATTRIBUTE>
```

```
<ATTRIBUTE>  
<TYPE>Token</TYPE>  
<FEATURE>length</FEATURE>  
<DATATYPE>numeric</DATATYPE>  
</ATTRIBUTE>
```

```
<ATTRIBUTE>  
<TYPE>Token</TYPE>  
<FEATURE>orth</FEATURE>  
<DATATYPE>nominal</DATATYPE>  
</ATTRIBUTE>
```

```
<ATTRIBUTE>  
<TYPE>Token</TYPE>  
<FEATURE>string</FEATURE>  
<DATATYPE>nominal</DATATYPE>  
</ATTRIBUTE>
```

```
</ML-CONFIG>
```

- For this task, we are using attribute features
- These allow us to take features from the instance annotations or others that are co-located with them
- We specify type, feature and datatype
- Attribute features also can be taken from instances nearby
- That's a bit less useful with a sequence learner though—why?



GATE Developer 8.5-SNAPSHOT build ee930e5

File Options Tools Help

Messages ft-bt-wireless... ANNIE

Loaded Processing resources

Name	Type
ANNIE NE Transducer	ANNIE NE Transducer
ANNIE OrthoMatcher	ANNIE OrthoMatcher

Selected Processing resources

Name	Type
Document Reset PR	Document Reset
ANNIE English Tokeniser	ANNIE English To
ANNIE Gazetteer	ANNIE Gazetteer
ANNIE Sentence Splitter	ANNIE Sentence
ANNIE POS Tagger	ANNIE POS Tagge
Annotation Set Transfer 0009F	Annotation Set T
LF_TrainChunking 000A0	LF_TrainChunkin

Run "LF_TrainChunking 000A0"?

Corpus: training

Runtime Parameters for the "LF_TrainChunking 000A0" LF_TrainChunking:

Name	Type	Required	Value
algorithmParameters	String		
classAnnotationTypes	List	✓	[Person, Organization]
dataDirectory	URL	✓	file:/home/genevieve/svn/sale/talks/gate-course-jun18/module-5-ml/chunki
featureSpecURL	URL	✓	file:/home/genevieve/svn/sale/talks/gate-course-jun18/module-5-ml/chunki
inputASName	String		
instanceType	String	✓	Token
scaleFeatures	ScalingMethod	✓	NONE
seqEncoder	SeqEncoderEnum		BIO
sequenceSpan	String		Sentence
trainingAlgorithm	AlgorithmClassification		MalletCRF_SEQ_MR

Run this Application

Serial Application Editor Initialisation Parameters About...

Resource Features

gate.app.Metadata

gate.gui.icon

ANNIE run in 0.485 seconds

- Make sure you have selected the training corpus
- Run the application!

Chunking application parameters

- Now switch off the training PR and create and add the chunking application PR
- (You can switch off the annotation set transfer too)
- It doesn't have a targetFeature parameter like the classification application PR did
- You don't need to tell it what type to create because the model knows it from training!
- Set dataDirectory to the location where you told the training PR to put the model
- Set the sequence span



Applying

The screenshot shows the GATE Developer interface with the following components:

- Messages:** ANNIE
- Loaded Processing resources:**

Name	Type
ANNIE NE Transducer	ANNIE NE Transducer
ANNIE OrthoMatcher	ANNIE OrthoMatcher
Annotation Set Transfer 00036	Annotation Set Trans
LF_TrainChunking 00030	LF_TrainChunking
- Selected Processing resources:**

Name	Type
Document Reset PR	Document Reset PR
ANNIE English Tokeniser	ANNIE English Tokeniser
ANNIE Gazetteer	ANNIE Gazetteer
ANNIE Sentence Splitter	ANNIE Sentence Splitter
ANNIE POS Tagger	ANNIE POS Tagger
LF_ApplyChunking 00031	LF_ApplyChunking
- Run "LF_ApplyChunking 00031"?:** Yes No If value of feature is
- Corpus:** training
- Runtime Parameters for the "LF_ApplyChunking 00031" LF_ApplyChunking:**

Name	Type	Required	Value
algorithmParameters	String		
confidenceThreshold	Double	✓	0.0
dataDirectory	URL	✓	file:/home/genevieve/svn/sale/talks/gate-course-jun16/module-3-ml-barbour/chunkir
inputASName	String		
instanceType	String	✓	Token
outputASName	String		LearningFramework
sequenceSpan	String		Sentence
- Buttons:** Run this Application, Serial Application Editor, Initialisation Parameters, About...
- Status:** ANNIE run in 14.256 seconds

- Now run this on the test corpus

Chunking—Evaluation using Corpus QA

Chunking Evaluation

- For classification, each response is simply right or wrong
- For NER, there are more ways to be wrong
 - Fewer or more mentions than there really are, or you can overlap
- So we need different metrics

What are precision, recall and F1?

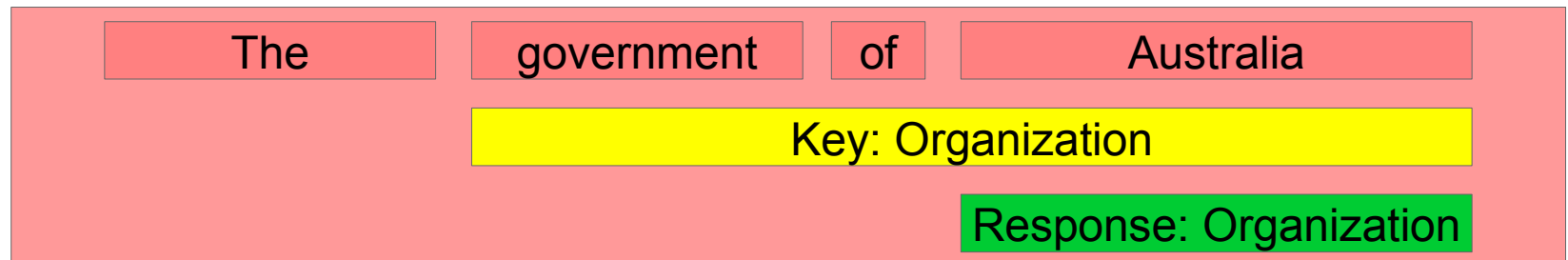
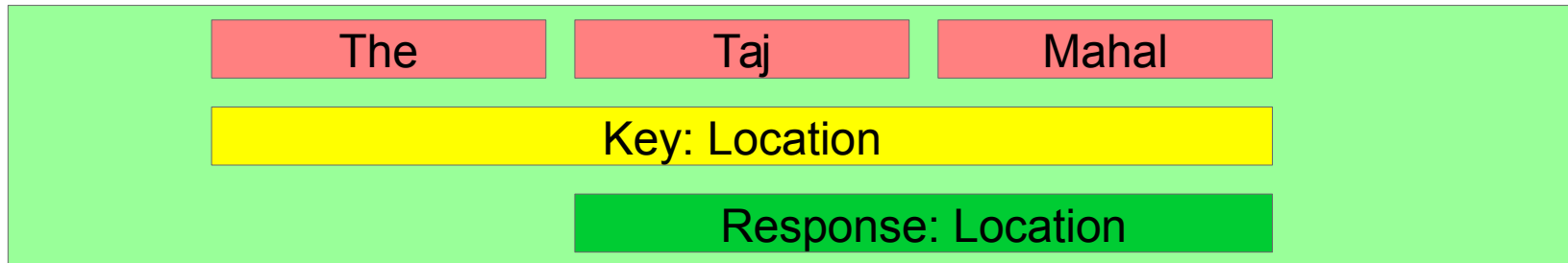
- Precision: what proportion of our automatic annotations were correct?
- Recall: what proportion of the correct annotations did our automatic tool create?
- $P = \text{correct} / (\text{correct} + \text{spurious}) = \text{tp} / (\text{tp} + \text{fp})$
- $R = \text{correct} / (\text{correct} + \text{missing}) = \text{tp} / (\text{tp} + \text{fn})$
- where tp = true positives, fp = false positives, fn = false negatives

What are precision, recall and F1?

- F-score is an amalgam of the two measures
- $F_{\beta} = (1+\beta^2)PR / (\beta^2 P + R)$
 - The equally balanced F1 ($\beta = 1$) is the most common F-measure
 - $F1 = 2PR / (P + R)$

Strict and Lenient

- “Strict” means we count an annotation as correct only if it has the same span as the gold standard annotation
- Lenient means we allow an annotation that overlaps to be correct, even if it isn't a perfect span match
- Which do you think is the right way to do it?





Examining the results of application

The screenshot shows the GATE Developer 8.5-ANNIE interface. The main window displays a document with several paragraphs of text. Annotations are visible on the text, including 'Mr Beresford' (Person), 'GKN' (Organization), 'Toyota Camry' (Organization), and 'AugustaWestland' (Organization). The 'Annotations Stack' tab is active, showing a list of annotation sets: Lookup, Sentence, SpaceToken, Split, Token, Key, Date, Location, Money, Organization, Percent, Person, LF_SEQ_TMP, and LearningFramework. The 'LearningFramework' section is expanded, showing 'Organization' and 'Person' as selected. The 'Document Editor' tab is also visible, showing the text 'yota to supply driveline components for the Toyota Camry. Increased sales in' with colored boxes indicating annotations.

Examine a document from the test corpus

You should have a new "LearningFramework" AS with Person and Organization annotations

The original annotations (in the Key AS) are similar but not always identical!

The Annotations Stack is good for comparing them

How similar do they appear to be? Do you think you will get a good result?

Comparing the Sets with Corpus QA



GATE Developer 8.5-SNAPSHOT build ee930e5

File Options Tools Help

Messages ANNIE test ft-BT-loop-01-a... ft-GKN-09-aug-2...

Corpus statistics Document statistics

Annotation	Match	Only A	Only B	Overlap	Prec. B/A	Rec. B/A	F1.0-a.
Organization	523	147	108	43	0.8079	0.7637	0.7851
Person	149	40	31	4	0.8207	0.7824	0.8011
Macro summary					0.8143	0.7731	0.7931
Micro summary	672	187	139	47	0.8106	0.7677	0.7885

Annotation Sets A/Key & B/Response

[Default set]

Key (A)

LearningFramework (B)

LF_SEQ_TMP

present in every document

Annotation Types

Date

Location

Money

Organization

present in every selected set

Annotation Features

1

gender

kind

LF_confidence

present in every selected type

Measures Options

F-Score Classification

F1.0-score strict

F1.0-score lenient

F1.0-score average

F1.0-score strict BDM

Compare

Resource Features

Corpus editor Initialisation Parameters Corpus Quality Assurance

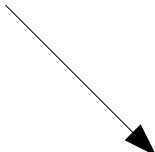
Views built!

- Select the test corpus and click on the Corpus Quality Assurance tab (it will take a few seconds to scan the documents)
- Select the Key and LearningFramework annotation sets as A and B, respectively
- Select the “Person” type
- Choose an F-measure
- Click on Compare
- Did you get a good result?



Using Annotation Diff to examine performance

Change type here



Annotation Difference

Key doc: ft-claims-direct-10-a... Key set: Key Type: Person Weight: Compare

Resp. doc: ft-claims-direct-10-a... Resp. set: LearningFra... Features: all some none 1.0

Start	End	Key	Features	=?	Start	End	Response	Featur
1549	1557	Mr-Poole	{rule=PersonFinal, g...l=PersonTitleGender}	=	1549	1557	Mr-Poole	{LF_confidence=0.857
1534	1544	Mr-Sullman	{rule=PersonFinal, g...l=PersonTitleGender}	=	1534	1544	Mr-Sullman	{LF_confidence=0.804
1201	1211	Mr-Sullman	{rule=PersonFinal, g...l=PersonTitleGender}	=	1201	1211	Mr-Sullman	{LF_confidence=0.850
1188	1196	Mr-Poole	{rule=PersonFinal, g...l=PersonTitleGender}	=	1188	1196	Mr-Poole	{LF_confidence=0.848
916	924	Mr-Poole	{rule=PersonFinal, g...l=PersonTitleGender}	=	916	924	Mr-Poole	{LF_confidence=0.848
901	911	Mr-Sullman	{rule=PersonFinal, g...l=PersonTitleGender}	=	901	911	Mr-Sullman	{LF_confidence=0.842
710	721	Colin-Poole	{rule=PersonFinal, g...e, rule1=PersonFull}	=	710	721	Colin-Poole	{LF_confidence=0.545
809	824	Simon-Ware-Lane	{}	?				
693	705	Tony-Sullman	{rule=PersonFinal, g...e, rule1=PersonFull}	?				
1822	1829	Sullman	{}	?				
1834	1839	Poole	{}	?				
				?	2569	2582	Claims-Direct	{LF_confidence=0.587
				?	2073	2083	High-Court	{LF_confidence=0.536
				?	2173	2186	Claims-Direct	{LF_confidence=0.476
				?	602	615	Claims-Direct	{LF_confidence=0.628
				?	0	13	Claims-Direct	{LF_confidence=0.677

16 pairings have been found (0 annotations are hidden)

Correct:	7	Recall	Precision	F-measure	
Partially correct:	0	Strict:	0.64	0.58	0.61
Missing:	4	Lenient:	0.64	0.58	0.61
False positives:	5	Average:	0.64	0.58	0.61

Statistics Adjudication

Switch to the "Document statistics" tab

Choose a document

Click on the Annotation Diff icon



What kind of mistakes did your application make?

Using Annotation Diff...

- “Correct”: the response annotation has the right feature and span
- “Partially correct”: response has the right feature and overlapping but not exactly matched span; this counts as correct in the “lenient” scoring
- “Missing”: key annotation+feature is missing from the response (a.k.a. “false negative”)
- “False positive”: response annotation+feature shouldn't be there (a.k.a. “spurious”)



Classification Evaluation PR for Chunking?

- We didn't use a Learning Framework evaluation PR for this chunking task
- What do you think would happen if you used the Classification Evaluation PR to do a chunking problem?
- It would work! It would evaluate the accuracy of the system in correctly identifying beginnings, insides and outsides
- However, it wouldn't tell you much about how well you did finding named entities
 - There are so many outsides that you can get a high score just by saying everything is an outside!
- You could use it to tune parameters if you wanted, though



Exercise—Improving the result

- Again, see if you can improve your result
- Try different features and algorithms



Exercise 2

- Try to learn different entity types

Exporting Feature Data

Exporting feature data

- A GATE ML PR serves a number of functions
 - Scraping features off the documents and formulating them as ML training sets
 - Sending the training sets to ML libraries to train a model
 - Creating instances (without class) at apply time to send to a trained model to be classified and writing the resulting class back onto the application instance
- We have integrated quite a few algorithms and some ML facilitation technology, so many ML tasks can be accomplished entirely in GATE

Exporting feature data

- However, GATE isn't an ML tool—its forte and contribution is complex linguistic features. There is a limit to what we will include in the way of ML innovations.
- For example, the Learning Framework;
 - doesn't include feature selection technologies
 - includes only limited feature scaling
 - doesn't integrate all algorithm variants



Exporting feature data

- For more advanced needs, there are other ways to work
- You can export your training set and use it to train a model outside of GATE
 - The Learning Framework will allow you to use a model trained outside of GATE to create an application
- Exporting data and working in e.g. Weka can also provide a faster way to tune parameters
 - When you change parameters in the LF it starts over again scraping the features off the documents, which takes a long time on a big corpus
- You could use e.g. Weka's feature selection technology and then recreate what you learned back into GATE by editing your feature spec
- It can also be a good sanity check to see your data in export format



Export the data as ARFF

- Create an Export PR and add it to the application
- You can turn off the other Learning Framework PRs
- Annotation Set Transfer needs to stay though

Export Parameters

- `classAnnotationTypes` is as for training, and its presence indicates that we are exporting a CHUNKING dataset—set it to `Person` and `Organization`
- `dataDirectory`, `featureSpecURL`, `inputASName` and `instanceType` you are familiar with by now—set them
- For exporter, choose `ARFF_CL_MR*`
- Don't set target feature! This would indicate that we want to export a classification dataset!
- Don't set `sequenceSpan`—this would indicate that we want to export data in a format suitable for training a sequence learner. This isn't supported yet.

* "CL" means classification—why are we exporting a classification dataset for a chunking problem? Because they're all classification behind the scenes. GATE turns the chunking problem into a classification problem for training and then turns it back again!

GATE Developer 8.2-SNAPSHOT build 5490

File Options Tools Help

Messages ANNIE test

Language Resources

- in-tesco-citywire-07
- in-scoot-10-aug-200
- in-reed-10-aug-2001
- in-outlook-10-aug-2
- in-oil-09-aug-2001.x
- in-german-bank-10-
- in-bayer-10-aug-200
- in-airlines-08-aug-20
- in-GKN-citywire-10-
- gu-w&d-10-aug-200
- gu-telewest-10-aug-
- gu-synergie-10-aug-
- gu-singtel-10-aug-2
- gu-scoot-10-aug-20
- gu-ryanair.xml_0008
- gu-recession-6-aug-
- gu-manuf-jobs-07-a
- qu-m&s-10-aug-200

Loaded Processing resources

Name	Type
ANNIE NE Transducer	ANNIE NE Transducer
ANNIE OrthoMatcher	ANNIE OrthoMatcher
LF_ApplyChunking 00031	LF_ApplyChunking
LF_TrainChunking 00030	LF_TrainChunking

Selected Processing resources

Name	Type
Document Reset PR	Document Reset
ANNIE English Tokeniser	ANNIE English Tok
ANNIE Gazetteer	ANNIE Gazetteer
ANNIE Sentence Splitter	ANNIE Sentence S
ANNIE POS Tagger	ANNIE POS Tagge
Annotation Set Transfer 00036	Annotation Set Tr
LF_Export 00099	LF_Export

Run "LF_Export 00099"?

Yes No If value of feature is

Corpus: test

Runtime Parameters for the "LF_Export 00099" LF_Export:

Name	Type	Required	Value
algorithmParameters	String		
classAnnotationType	String		Person
dataDirectory	URL	✓	file:/home/genevieve/svn/sale/talks/gate-course-jun16/module-3-ml-barbour
exporter	Exporter	✓	EXPORTER_ARFF_CLASS
featureSpecURL	URL	✓	file:/home/genevieve/svn/sale/talks/gate-course-jun16/module-3-ml-barbour
inputASName	String		
instanceType	String	✓	Token
scaleFeatures	ScalingMethod	✓	NONE
sequenceSpan	String		
targetFeature	String		
targetType	TargetType	✓	NOMINAL

Run this Application

Serial Application Editor Initialisation Parameters About...

ANNIE run in 2.811 seconds

- Set targetType to nominal, because beginnings, insides and outsides are nominal classes
- Go ahead and export the data!

Examining the ARFF



```
data.arff (~/.svn/sale/talks/gate-course-jun16/module-3-ml/chunking-hands-on) - gedit
File Edit View Search Tools Documents Help
data.arff x
@attribute A:Token:string=eyes numeric
@attribute A:Token:string=gyms numeric
@attribute A:Token:string=contributes numeric
@attribute A:Token:string=Like-for-like numeric
@attribute A:Token:string=645 numeric
@attribute A:Token:string=Separately numeric
@attribute A:Token:string=small-cap numeric
@attribute A:Token:string=Espress numeric
@attribute A:Token:string=Top numeric
@attribute A:Token:string=Notch numeric
@attribute class {0,B,I}

@data
{0 1,1 1,2 4,3 1,4 1}
{1 1,2 2,5 1}
{1 1,2 7,3 1}
{1 1,2 18,3 1}
{1 1,2 7}
{1 1,2 3}
{1 1,2 8}
{1 1,2 3}
{1 1,2 6}
{2 1}
{1 1,2 1}
{1 1,2 4}
{1 1,2 2,5 1}
Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

- You'll find your exported ARFF in your dataDirectory, called data.arff
- At the top are a list of attributes. Are they as expected?
- The last attribute is the class attribute. Do you see it?
- After that come feature vectors in sparse format. How can you tell that they are in sparse format? What would this file look like if they were written out in full?



Wrapper projects

- In this session we've only used fully integrated algorithms, to get us started
- But most of the libraries are integrated as wrappers, for licensing reasons
- You can find out how to use these libraries by consulting the documentation—it isn't hard!
- For example, deep learning (neural net) libraries are integrated in this way.

`https://gatenlp.github.io/gateplugin-LearningFramework/`