



---

# Module 11: Machine Learning





# Module 11 Outline

09:45–11:00

- What is machine learning and why do we want to do it?
- Setting up a corpus
- The ML configuration file

11:00–11:15

BREAK

11:15–12:30

- Running the ML PR in evaluation mode
- Evaluation in ML
- Training, application, corpus QA

12:30–14:00

LUNCH

14:00–16:00

- Varying the configuration file
- Engines & algorithms

16:00–16:30

BREAK

16:30–17:30

TALK



---

# What is Machine Learning and why do we want to do it?



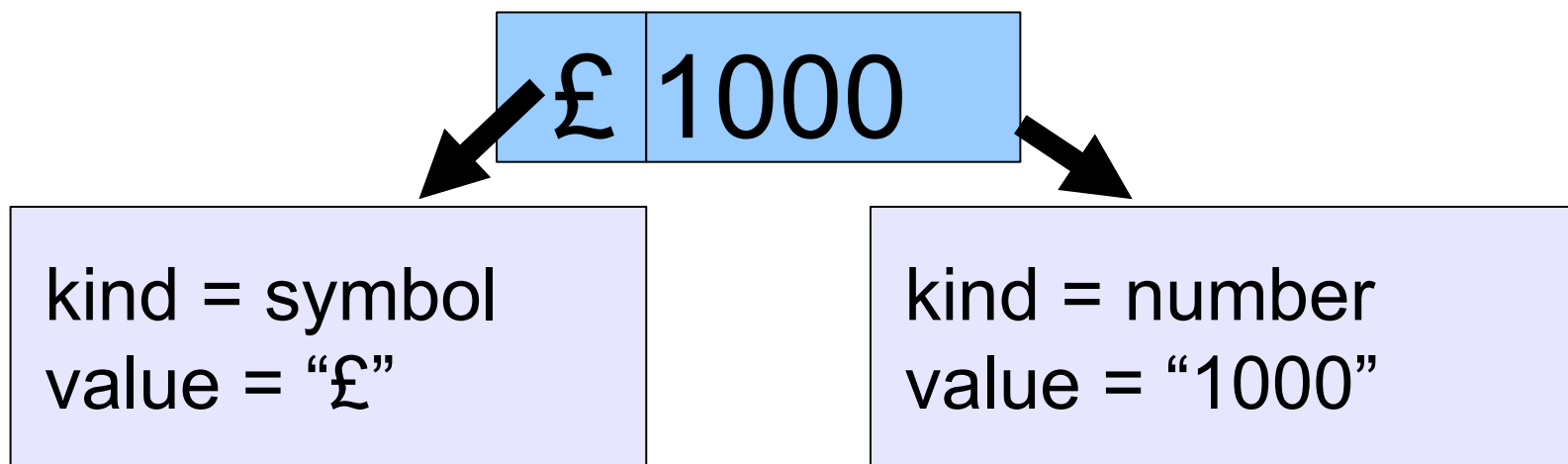
# What is ML?

---

- Aim to automate the process of inferring new data from existing data
- In GATE, that means creating annotations by learning how they relate to other annotations

# Learning a pattern

- For example, we have “Token” annotations with “kind” and “value” features



- ML could learn that a “£” followed by a number is an amount of currency

# How is that better than **GATE** making rules (e.g. JAPE)?

---

- It is different to the rule-based approach
- Some things humans are better at writing rules for, and some things ML algorithms are better at finding
- With ML you don't have to create all the rules
- However you do have to manually annotate a training corpus
- Rule-based approaches (e.g. JAPE) and ML work well together
  - e.g. JAPE often used extensively to prepare data for ML



# Terminology: Instances, attributes, classes

California Governor Arnold Schwarzenegger proposes deep cuts.

**Instances:**

Any annotation  
Tokens are often convenient



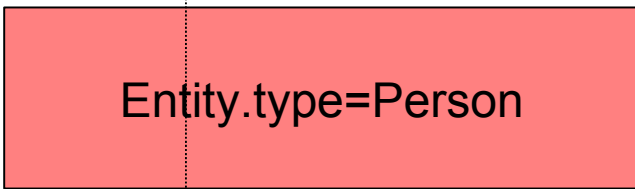
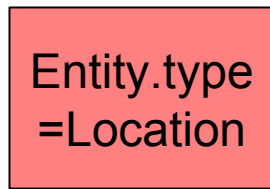
**Attributes:**

Any annotation feature relative to instances  
Token.String  
Token.category (POS)  
Sentence.length



**Class:**

The thing we want to learn  
A feature on an annotation





# 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?
  - `Token.string == "California" → Location`



# Attributes

---

- Attributes are pieces of information about instances
- They are sometimes called “features” in machine learning literature
- Examples
  - `Token.string == “Arnold”`
  - `Token.orth == upperInitial`
  - `Token(-1).string == “Governor”`



# Classes

---

- The class is what we want to learn
- For example, if we want to find person names, for every instance, the question is, is this a person name?
  - The classes are “yes” and “no”
- Sometimes there are many classes, for example we may want to learn entity types
  - For every instance, the question is, which of a predetermined entity type set does this belong to?



# ML Tasks

---

GATE supports 3 types of ML tasks:

- chunk recognition (named entity recognition, NP chunking)
- text classification (sentiment classification, POS tagging)
- relation annotation



# Training

---

- Training involves presenting data to the ML algorithm from which it creates a model
- The training data (instances) have been annotated with class annotations as well as attributes
- Models are representations of decision-making processes that allow the machine learner to decide what class the instance has based on the attributes of the instance



# Application

---

- When the machine learner 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
- The test corpus must not be the same corpus as you trained on
  - This would give the machine learner an advantage, and would give a false idea of how good it is
- GATE's ML PR has a built-in evaluation mode that splits the corpus into training and test sets and cross-validates them



---

# Setting up a Corpus



# Load the corpus

---

- Create a corpus (any name is fine)
- Populate it from `module-11-hands-on/corpus/*.xml` in your hands-on materials
- Use UTF-8 encoding
- 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
- There are also some annotations in the “Original markups” set



# What are we going to use this corpus for?

---

- We are going to train a machine learner to annotate corpora with these entity types
- We need a training corpus and a test corpus
- The training corpus will be used by the machine learner to deduce relationships between attributes and entity types (classes)
- The test corpus will be used to find out how well it is working, by comparing annotations created by the learner with the correct annotations that are already there
- In *Evaluation* mode, which we will try first, the ML PR automatically splits one corpus up into training and test sets

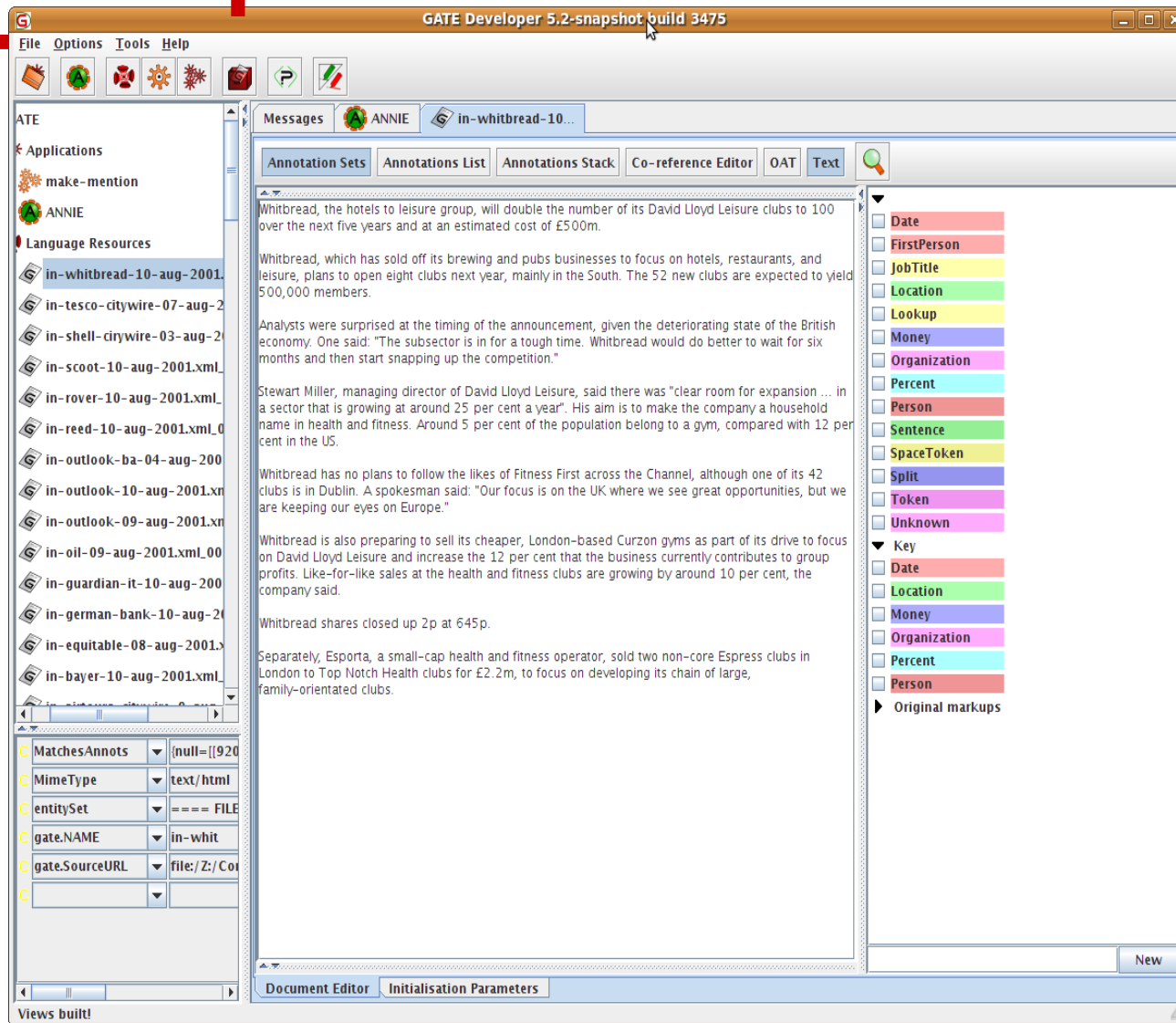


# Instances and Attributes

---

- This corpus so far contains only the class annotations
- There is not much in this corpus to learn from
- What would our instances be?
- What would our attributes be?
- If we run ANNIE over the corpus, then we can use “Token” annotations for instances, and we would have various options for attributes
- **Load ANNIE but add the Key AS to setsToKeep in the document reset PR!**
- **Run ANNIE over your corpus**

# Running ANNIE on the corpus

The screenshot shows the GATE Developer 5.2-snapshot build 3475 interface. The main window displays a document titled "in-whitbread-10..." with text from a news article. The text is annotated with various entities and relations. The left sidebar shows the "Applications" list with "ANNIE" selected. The right sidebar shows the "Annotation Sets" list with "Date", "FirstPerson", "JobTitle", "Location", "Lookup", "Money", "Organization", "Percent", "Person", "Sentence", "SpaceToken", "Split", "Token", "Unknown", "Key", "Date", "Location", "Money", "Organization", "Percent", "Person", and "Original markups" listed. The bottom status bar shows "Document Editor" and "Initialisation Parameters".

Messages

ANNIE in-whitbread-10...

Annotation Sets Annotations List Annotations Stack Co-reference Editor OAT Text

Whitbread, the hotels to leisure group, will double the number of its David Lloyd Leisure clubs to 100 over the next five years and at an estimated cost of £500m.

Whitbread, which has sold off its brewing and pubs businesses to focus on hotels, restaurants, and leisure, plans to open eight clubs next year, mainly in the South. The 52 new clubs are expected to yield 500,000 members.

Analysts were surprised at the timing of the announcement, given the deteriorating state of the British economy. One said: "The subsector is in for a tough time. Whitbread would do better to wait for six months and then start snapping up the competition."

Stewart Miller, managing director of David Lloyd Leisure, said there was "clear room for expansion ... in a sector that is growing at around 25 per cent a year". His aim is to make the company a household name in health and fitness. Around 5 per cent of the population belong to a gym, compared with 12 per cent in the US.

Whitbread has no plans to follow the likes of Fitness First across the Channel, although one of its 42 clubs is in Dublin. A spokesman said: "Our focus is on the UK where we see great opportunities, but we are keeping our eyes on Europe."

Whitbread is also preparing to sell its cheaper, London-based Curzon gyms as part of its drive to focus on David Lloyd Leisure and increase the 12 per cent that the business currently contributes to group profits. Like-for-like sales at the health and fitness clubs are growing by around 10 per cent, the company said.

Whitbread shares closed up 2p at 645p.

Separately, Esporta, a small-cap health and fitness operator, sold two non-core Espress clubs in London to Top Notch Health clubs for £2.2m, to focus on developing its chain of large, family-orientated clubs.

MatchesAnnots (null=[1920])

MimeType text/html

entitySet FILE

gate.NAME in-whit

gate.SourceURL file:/Z:/Co

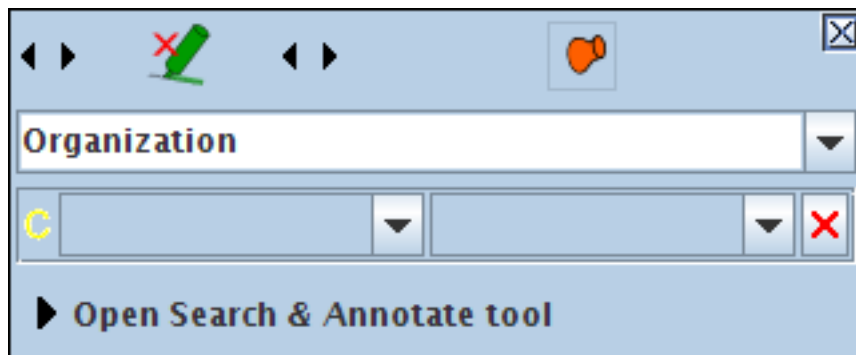
Document Editor Initialisation Parameters

Views built!

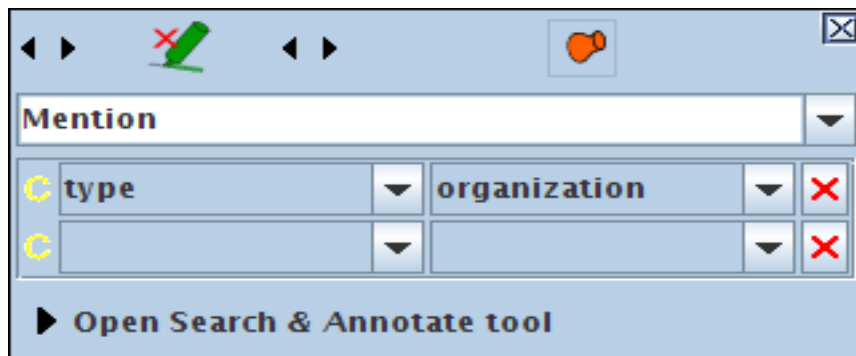
- Having run ANNIE on the corpus, we have more annotations to work with

# Preparing the corpus: Classes

- What we have:



- What we need:



# Preparing the corpus: Classes

---



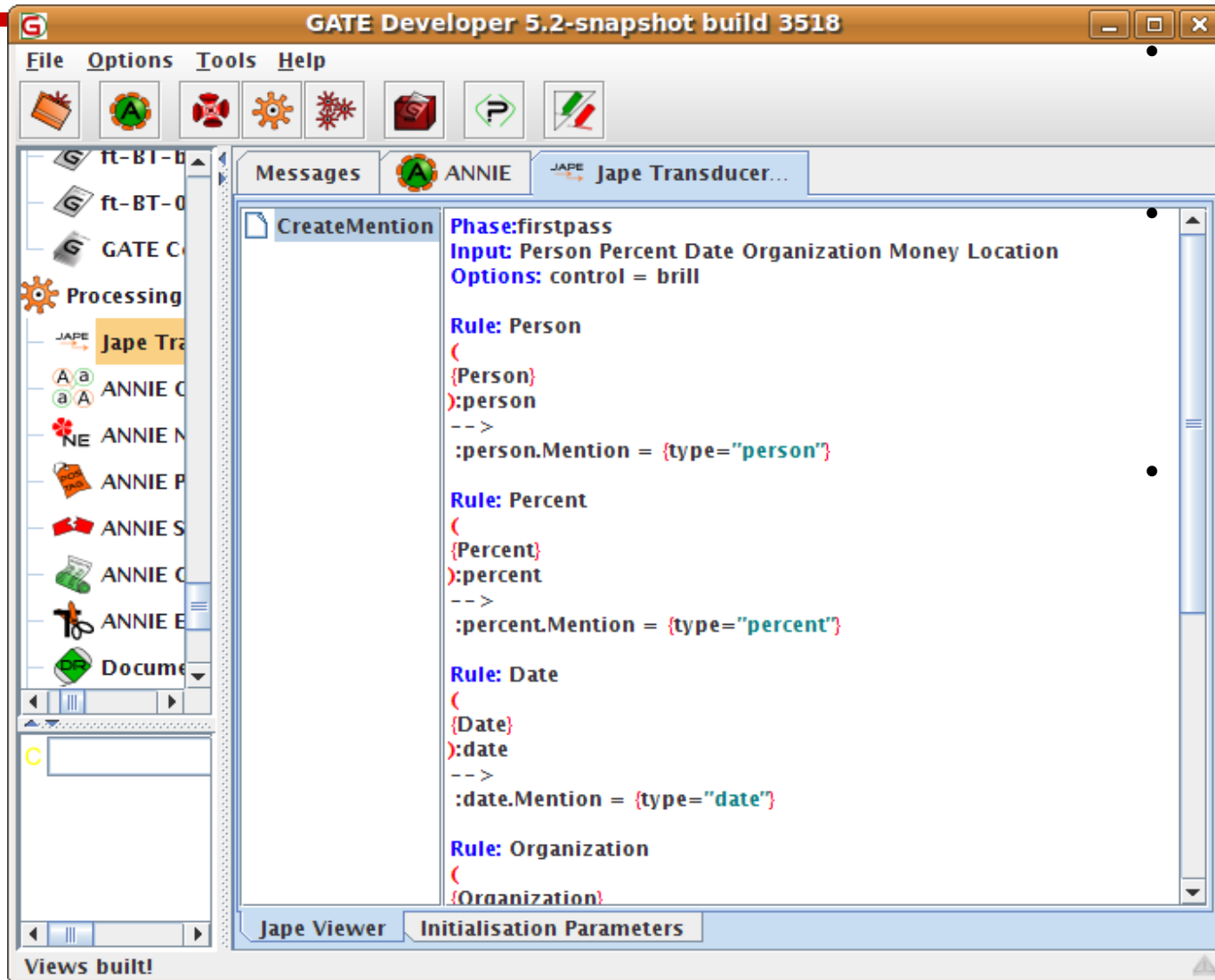
- Currently each class has its own annotation type (Date, Person, Percent etc.)
- However ML PR expects class to be a feature value, not a type
- Therefore we are going to make a new annotation type for the ML to learn from, e.g. “Mention”, though it does not matter what it is called



# Making class annotations

- 
- Load a JAPE transducer from the [module-11-hands-on/CreateMention.jape](#) grammar
  - Look at the grammar in GATE

# The CreateMention.jape grammar

GATE Developer 5.2-snapshot build 3518

File Options Tools Help

Messages ANNIE Jape Transducer...

CreateMention

```

Phase: firstpass
Input: Person Percent Date Organization Money Location
Options: control = brill

Rule: Person
(
{Person}
):person
-->
:person.Mention = {type="person"}

Rule: Percent
(
{Percent}
):percent
-->
:percent.Mention = {type="percent"}

Rule: Date
(
{Date}
):date
-->
:date.Mention = {type="date"}

Rule: Organization
(
{Organization}

```

Jape Viewer Initialisation Parameters

Views built!

This grammar makes a new annotation type called "Mention"

It makes the previous annotation type into a feature of the "Mention" annotation

- Feature name is "type" because "class" is reserved for ontologies

# Applying the grammar to **GATE** the corpus

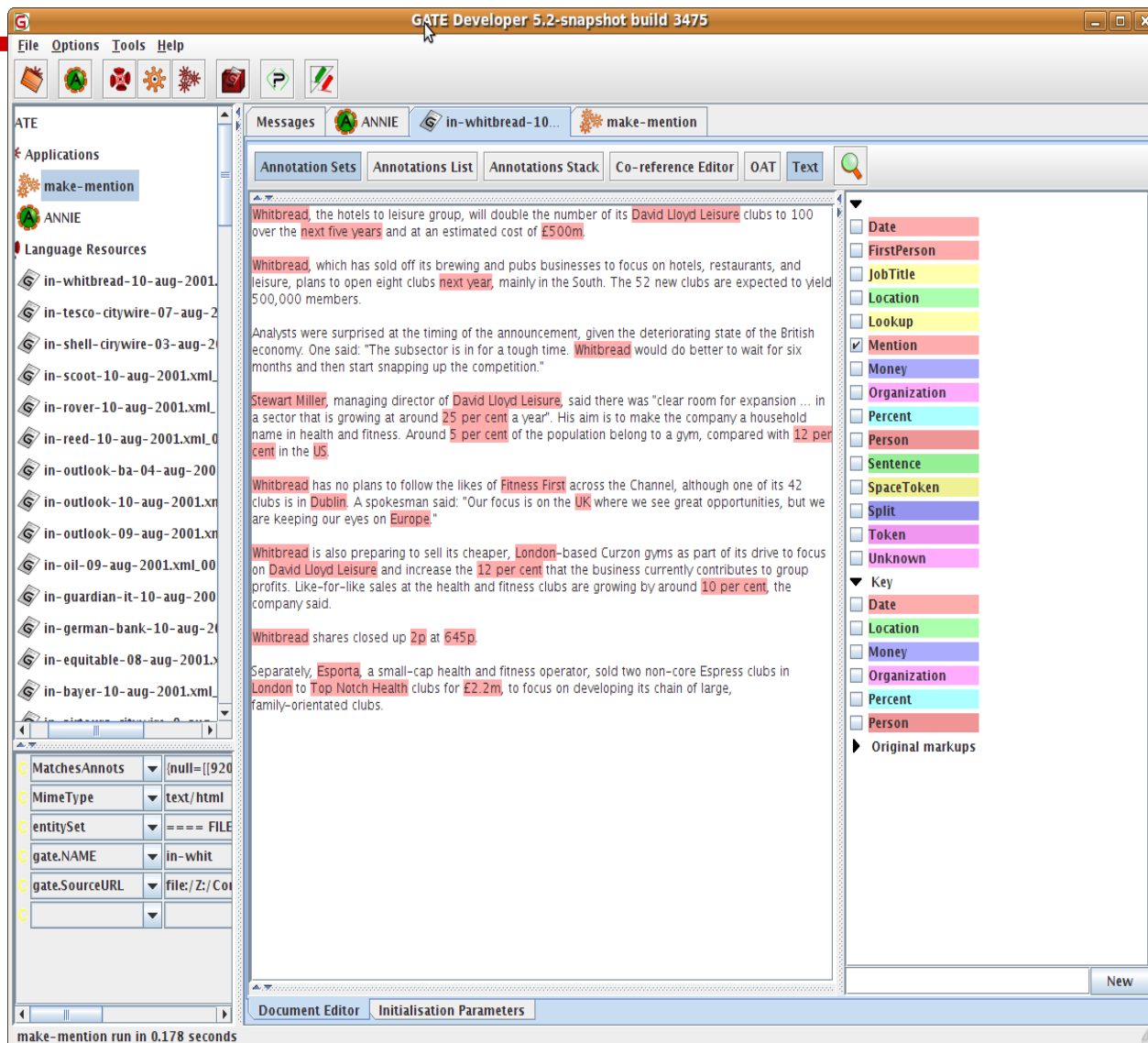
The screenshot shows the GATE Developer interface with the 'Jape Transducer' selected in the 'Processing Resources' tree. The 'Messages' tab is active, showing the 'ANNIE' application. The 'Loaded Processing resources' table contains one entry: 'Corpus Pipeline\_00056' of type 'Corpus Pipe'. The 'Selected Processing resources' table contains several entries, including 'Jape Transducer\_00094' which is highlighted. Below the tables, the 'Corpus' is set to 'GATE Corpus\_0001A'. The 'Runtime Parameters for the "Jape Transducer\_00094" Jape Transducer:' table is shown with the following data:

Name	Type	Required	Value
inputASName	String	Key	
ontology	Ontology	<none>	
outputASName	String		

At the bottom of the window, there is a 'Run this Application' button and a 'Serial Application Editor' tab. The status bar at the bottom left indicates 'loaded in 0.032 seconds'.

- Add the JAPE transducer at the end of your ANNIE application
- Set the inputASName to “Key”
- Leave the outputASName blank (default)

# Check the “Mention” annotations

GATE Developer 5.2-snapshot build 3475

File Options Tools Help

Messages ANNIE in-whitbread-10... make-mention

Annotations Sets Annotations List Annotations Stack Co-reference Editor OAT Text

Whitbread, the hotels to leisure group, will double the number of its David Lloyd Leisure clubs to 100 over the next five years and at an estimated cost of £500m.

Whitbread, which has sold off its brewing and pubs businesses to focus on hotels, restaurants, and leisure, plans to open eight clubs next year, mainly in the South. The 52 new clubs are expected to yield 500,000 members.

Analysts were surprised at the timing of the announcement, given the deteriorating state of the British economy. One said: "The subsector is in for a tough time. Whitbread would do better to wait for six months and then start snapping up the competition."

Stewart Miller, managing director of David Lloyd Leisure, said there was "clear room for expansion ... in a sector that is growing at around 25 per cent a year". His aim is to make the company a household name in health and fitness. Around 5 per cent of the population belong to a gym, compared with 12 per cent in the US.

Whitbread has no plans to follow the likes of Fitness First across the Channel, although one of its 42 clubs is in Dublin. A spokesman said: "Our focus is on the UK where we see great opportunities, but we are keeping our eyes on Europe."

Whitbread is also preparing to sell its cheaper, London-based Curzon gyms as part of its drive to focus on David Lloyd Leisure and increase the 12 per cent that the business currently contributes to group profits. Like-for-like sales at the health and fitness clubs are growing by around 10 per cent, the company said.

Whitbread shares closed up 2p at 645p.

Separately, Esporta, a small-cap health and fitness operator, sold two non-core Espress clubs in London to Top Notch Health clubs for £2.2m, to focus on developing its chain of large, family-orientated clubs.

Annotations List:

- Date
- FirstPerson
- JobTitle
- Location
- Lookup
- Mention
- Money
- Organization
- Percent
- Person
- Sentence
- SpaceToken
- Split
- Token
- Unknown
- ▼ Key
  - Date
  - Location
  - Money
  - Organization
  - Percent
  - Person
- ▶ Original markups

MatchesAnnots | null=[920  
 MimeType | text/html  
 entitySet | FILE  
 gate.NAME | in-whit  
 gate.SourceURL | file:/Z:/Co

Document Editor Initialisation Parameters

make-mention run in 0.178 seconds

- Rerun the application
- Check that you have some “Mention” annotations
- Check that they have a feature “type” and that the values look right



---

# The Configuration File



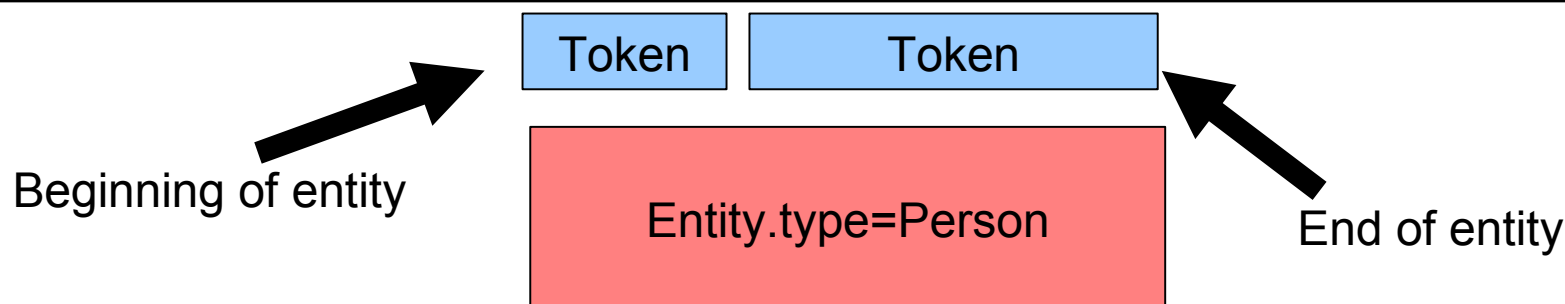
# Looking at the configuration file

---

- In the configuration file, we tell the machine learning PR what we want it to do
- You will find a configuration file in your hands-on materials, called ml-config-file.xml
- **Open it using a text editor**

<SURROUND value="true"/>

California Governor Arnold Schwarzenegger proposes deep cuts.



- This learned class covers more than one instance....
- Begin / End boundary learning
- Dealt with by API: *surround mode*
- Transparent to the user



# Confidence Thresholds

---

```
<PARAMETER name="thresholdProbabilityEntity" value="0.2"/>
```

```
<PARAMETER name="thresholdProbabilityBoundary" value="0.4"/>
```

- Learner will provide confidence ratings—how likely is a result to be correct
- We must determine how certain is good enough
- Depending on the application we might prefer to include or exclude annotations for which the learner is not too sure
- `thresholdProbabilityBoundary` is a threshold for the beginning and end instances
- `thresholdProbabilityEntity` is a threshold for beginning and end instances combined



```
<multiclassification2Binary  
method="one-vs-others"/>
```

California Governor Arnold Schwarzenegger proposes deep cuts.

Entity.type  
=Location

Entity.type=Person

- Many algorithms are binary classifiers (e.g. yes/no)
- We have several classes (Person, Location, Organization etc.)
- Therefore the problem must be converted so that we can use binary algorithms to solve it
- **one-vs-others**
  - LOC vs PERS+ORG / PERS vs LOC+ORG / ORG vs LOC+PERS
- **one-vs-another**
  - LOC vs PERS / LOC vs ORG / PERS vs ORG



```
<multiclassification2Binary  
method="one-vs-others"/>
```

- With more than a few classes, **one-vs-another** becomes very computationally expensive!
- **one-vs-others**:  $N$  classes  $\Rightarrow$   $N$  classifiers
  - A vs B+C+D, B vs A+C+D, C vs A+B+D, D vs A+B+C
- **one-vs-another**:  $N$  classes  $\Rightarrow$   $N \times (N+1) / 2$  classifiers
  - A vs B, A vs C, A vs D, B vs C, B vs D, C vs E

```
<EVALUATION method="holdout"  
ratio="0.66"/>
```



- 
- We are going to evaluate our application in two ways today
    - \_ The ML PR can automatically evaluate for us
    - \_ We will also run our own evaluation
  - This parameter dictates how the ML PR will evaluate for us, if we run it in evaluation mode
  - We are telling it that it should reserve a third of the data as a test set, train, then apply the result to the held out set
  - Alternatively, we could ask the PR to run a cross-validation evaluation



# Evaluation

---

`<EVALUATION method="kfold" runs="10"/>`

OR

`<EVALUATION method="holdout" ratio="0.66"/>`

- Holdout randomly picks ratio documents for training and the rest for testing; this is faster than k-fold because it only runs once
- But k-fold will give you more reliable results

# K-Fold Cross-Validation

- In k-fold cross-validation, the corpus is split into k equal parts, and the learner is trained k times on k-1 parts and evaluated on 1; the results are averaged
- For example, if k=4, the documents are split into groups A, B, C, & D, then:
  - \_ train on A+B+C, test on D
  - \_ train on A+B+D, test on C
  - \_ train on A+C+D, test on B
  - \_ train on B+C+D, test on A
  - \_ average these 4 results
- This maximises the training data without losing testing accuracy, but the Batch Learning PR takes 4 times as long
- Example:

```
<EVALUATION method="kfold" runs="10"/>
```



```
<ENGINE nickname="PAUM" ..
```

- 
- Next we specify what machine learning algorithm we wish to use
  - Today we are using the perceptron with uneven margins (“PAUM”)
  - We will use the following options:  
options="-p 50 -n 5 -optB 0.3"
    - Challenge: find out what these options do! (Hint: user guide §15.2)



# <INSTANCE- TYPE>Token</INSTANCE-TYPE>

---

- Next, we tell the ML PR what our instance annotation is
- The goal of the ML PR is, for every instance, to try to learn how the attributes of the instance relate to its class
- So the instance is a very critical concept
- We have decided that the “Token” is our instance annotation type
  - We made sure, earlier, that we have “Token” annotations in our corpus



# Specifying Attributes

```
<ATTRIBUTELIST>  
  <NAME>Form</NAME>  
  <SEMTYPE>NOMINAL</SEMTYPE>  
  <TYPE>Token</TYPE>  
  <FEATURE>category</FEATURE>  
  <RANGE from="-2" to="2"/>  
</ATTRIBUTELIST>
```

- For every attribute, we create a specification like the one above
- This is the information from which the PR will learn, so it is important to give it some good data
- You can see in the configuration file that there are several attributes, providing a good range of information
- However, if you have too many attributes it can take a very long time to learn!



# Breaking down the attribute specification

---

- `<NAME>Form</NAME>`
  - This is the name that we choose for this attribute. It can be anything we want, but it will help us later if we make it something sensible!
- `<SEMTYPE>NOMINAL</SEMTYPE>`
  - Is the value of this attribute a number or a name?



# Breaking down the attribute specification

---

- `<TYPE>Token</TYPE>`
  - The value of the attribute will be taken from the “Token” annotation
- `<FEATURE>category</FEATURE>`
  - The value of the attribute will be taken from the “category” feature



# Breaking down the attribute specification

---

```
<ATTRIBUTELIST>  
:  
  <RANGE from="-2" to="2"/>  
</ATTRIBUTELIST>
```

- Because this is an “ATTRIBUTELIST” specification, we can specify a “RANGE”
- In this case, we will gather attributes from the current instance and also the preceding and ensuing two



# Specifying the Class Attribute

```
<ATTRIBUTE>  
  <NAME>Class</NAME>  
  <SEMTYPE>NOMINAL</SEMTYPE>  
  <TYPE>Mention</TYPE>  
  <FEATURE>class</FEATURE>  
  <POSITION>0</POSITION>  
  <CLASS/>  
</ATTRIBUTE>
```

- You can call the class attribute whatever you want, but “Class” is a sensible choice
- Remember that our class attribute is the “class” feature of the “Mention” annotation
- This is an ATTRIBUTE, not an ATTRIBUTELIST, so we have “position”, not “range”
- The <CLASS/> element tells the BL PR that this is the class attribute to learn.

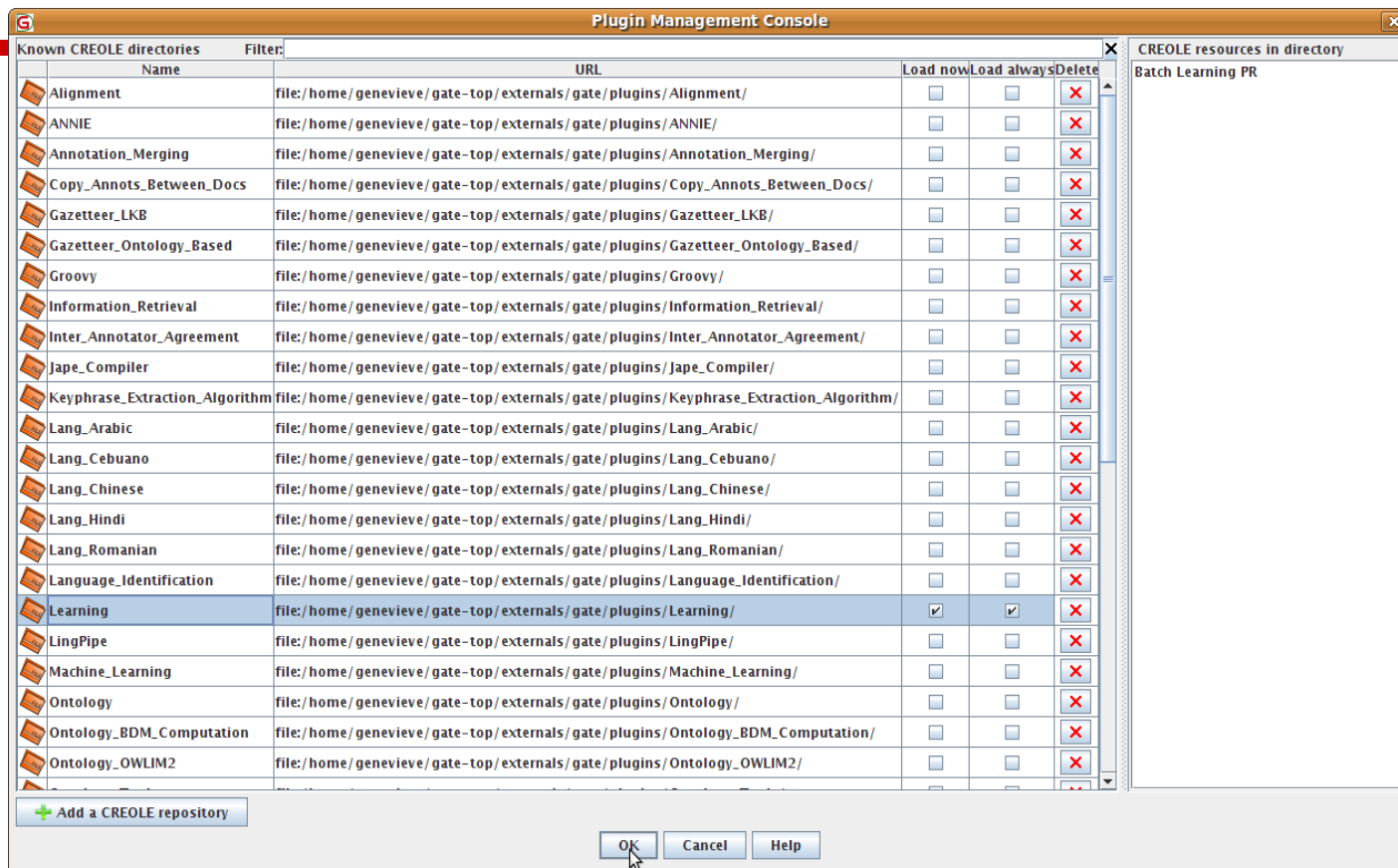


---

# Running the ML PR in evaluation mode



# Loading the Learning plugin



- Load the “Learning” plugin
- (We are **not** going to use the “Machine Learning” plugin, which is obsolete and does not have all the functionality we want.)

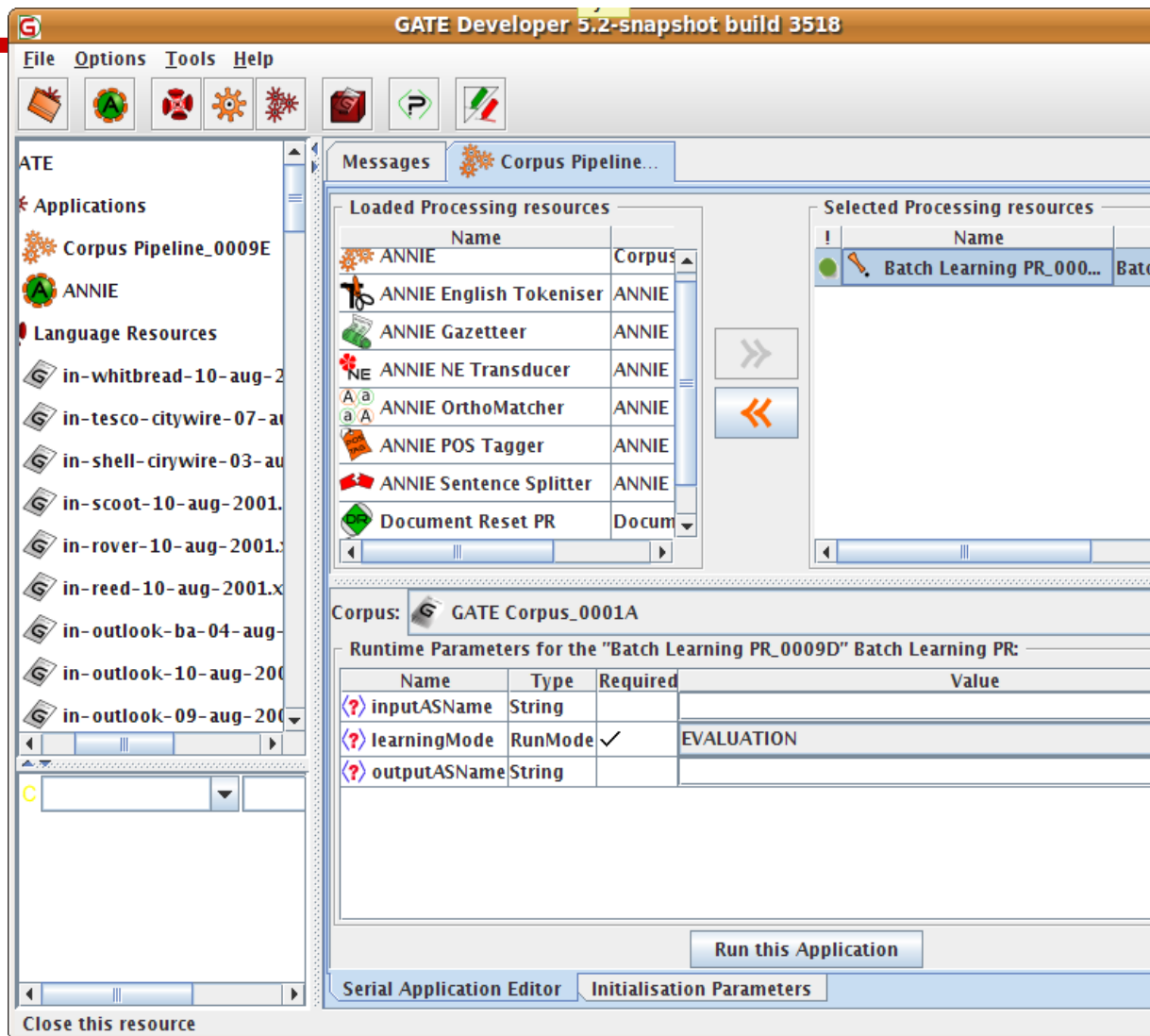


# Creating a Learning application

---

- **Create a “Batch Learning PR” using your configuration file**
- **Make a corpus pipeline application, and put the PR into it**

# Running the application in evaluation mode

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

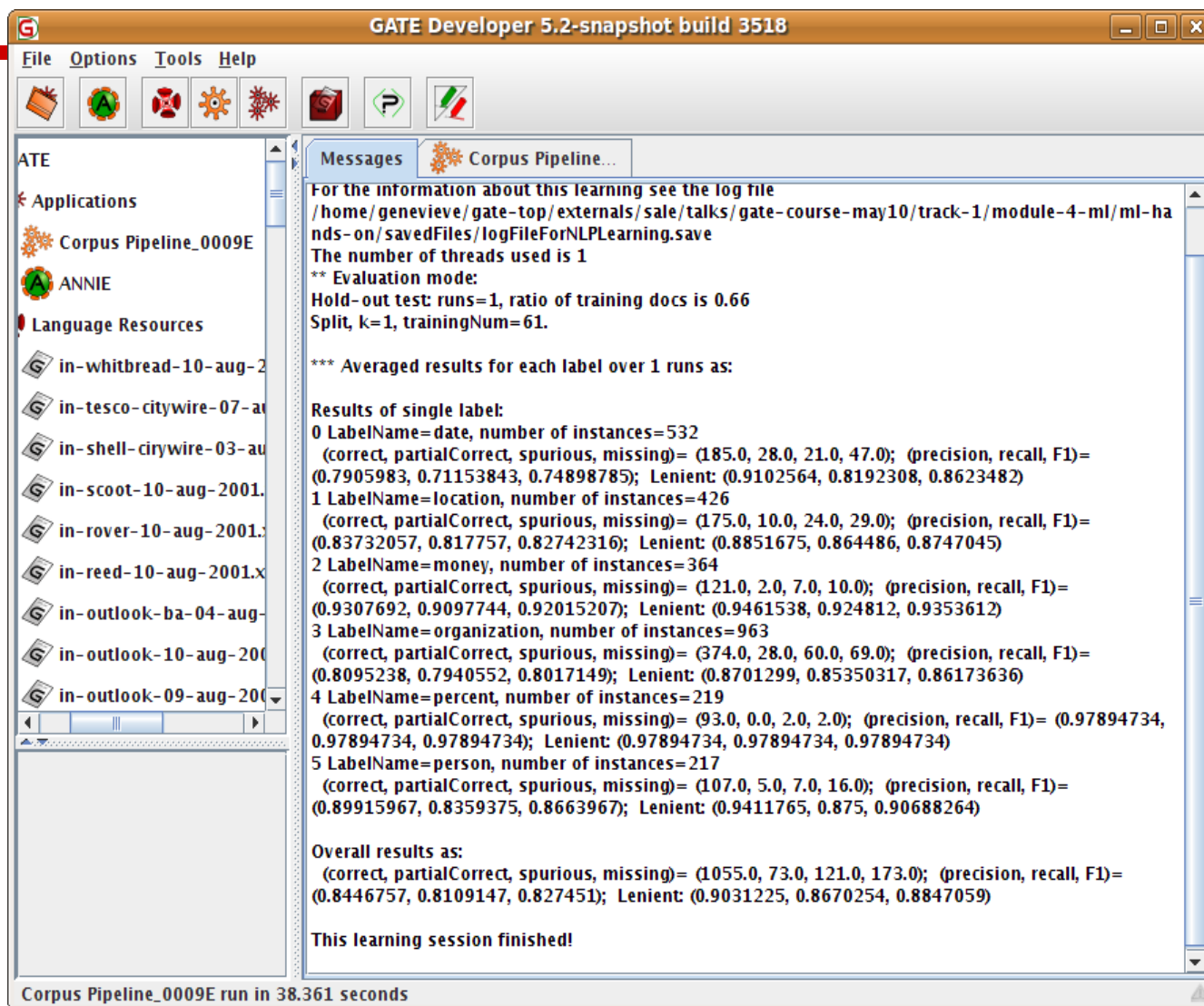
- Applications:** Corpus Pipeline\_0009E, ANNIE
- Language Resources:** in-whitbread-10-aug-2, in-tesco-citywire-07-a, in-shell-citywire-03-a, in-scoot-10-aug-2001, in-rover-10-aug-2001, in-reed-10-aug-2001.x, in-outlook-ba-04-aug, in-outlook-10-aug-20, in-outlook-09-aug-20
- Messages:** Corpus Pipeline...
- Loaded Processing resources:**

Name	Corpus
ANNIE	ANNIE
ANNIE English Tokeniser	ANNIE
ANNIE Gazetteer	ANNIE
ANNIE NE Transducer	ANNIE
ANNIE OrthoMatcher	ANNIE
ANNIE POS Tagger	ANNIE
ANNIE Sentence Splitter	ANNIE
Document Reset PR	Docum
- Selected Processing resources:** Batch Learning PR\_000... Batch
- Corpus:** GATE Corpus\_0001A
- Runtime Parameters for the "Batch Learning PR\_0009D" Batch Learning PR:**

Name	Type	Required	Value
inputASName	String		
learningMode	RunMode	✓	EVALUATION
outputASName	String		
- Buttons:** Run this Application
- Serial Application Editor:** Initialisation Parameters

- Make sure the corpus is selected
- The inputASName is blank because the attributes and class are in the default annotation set
- Select "EVALUATION" for the learningMode
- OutputASName should be the same as inputASName in evaluation mode
- Run the application!

# Inspecting the results



GATE Developer 5.2-snapshot build 3518

File Options Tools Help

Messages Corpus Pipeline...

For the information about this learning see the log file  
/home/genevieve/gate-top/externals/sale/talks/gate-course-may10/track-1/module-4-ml/ml-hands-on/savedFiles/logFileForNLPLearning.save  
The number of threads used is 1  
\*\* Evaluation mode:  
Hold-out test: runs=1, ratio of training docs is 0.66  
Split, k=1, trainingNum=61.

\*\*\* Averaged results for each label over 1 runs as:

Results of single label:

0 LabelName=date, number of instances=532  
(correct, partialCorrect, spurious, missing)= (185.0, 28.0, 21.0, 47.0); (precision, recall, F1)= (0.7905983, 0.71153843, 0.74898785); Lenient: (0.9102564, 0.8192308, 0.8623482)

1 LabelName=location, number of instances=426  
(correct, partialCorrect, spurious, missing)= (175.0, 10.0, 24.0, 29.0); (precision, recall, F1)= (0.83732057, 0.817757, 0.82742316); Lenient: (0.8851675, 0.864486, 0.8747045)

2 LabelName=money, number of instances=364  
(correct, partialCorrect, spurious, missing)= (121.0, 2.0, 7.0, 10.0); (precision, recall, F1)= (0.9307692, 0.9097744, 0.92015207); Lenient: (0.9461538, 0.924812, 0.9353612)

3 LabelName=organization, number of instances=963  
(correct, partialCorrect, spurious, missing)= (374.0, 28.0, 60.0, 69.0); (precision, recall, F1)= (0.8095238, 0.7940552, 0.8017149); Lenient: (0.8701299, 0.85350317, 0.86173636)

4 LabelName=percent, number of instances=219  
(correct, partialCorrect, spurious, missing)= (93.0, 0.0, 2.0, 2.0); (precision, recall, F1)= (0.97894734, 0.97894734, 0.97894734); Lenient: (0.97894734, 0.97894734, 0.97894734)

5 LabelName=person, number of instances=217  
(correct, partialCorrect, spurious, missing)= (107.0, 5.0, 7.0, 16.0); (precision, recall, F1)= (0.89915967, 0.8359375, 0.8663967); Lenient: (0.9411765, 0.875, 0.90688264)

Overall results as:  
(correct, partialCorrect, spurious, missing)= (1055.0, 73.0, 121.0, 173.0); (precision, recall, F1)= (0.8446757, 0.8109147, 0.827451); Lenient: (0.9031225, 0.8670254, 0.8847059)

This learning session finished!

Corpus Pipeline\_0009E run in 38.361 seconds

- The application may take a few minutes to run
- When it is finished, switch to the “Messages” tab to examine the results



# How well did we do?

---

- Here is my result:

**(precision, recall, F1)= (0.853012, 0.81629515, 0.83424973)**

- These figures look pretty good
- But what do they really mean?
- Next we will discuss evaluation measures
- Then we will run the PR in different modes
- Then we will see if we can get these numbers any higher!



---

# Evaluation in Machine Learning

# Recap of Evaluation in GATE



- Evaluation is an important part of information extraction work
  - We need to find out how good our application is by comparing its annotations to the “right answers” (manually prepared annotations)
  - Sometimes we need to compare annotations by different annotators, to see how consistent they are
- We use similar functions for both types of evaluation tasks



# Evaluation Mode

---

- We ran the machine learning PR in evaluation mode earlier
- We specified how the PR should run evaluation in the configuration file
- Once we had run the application, we obtained evaluation statistics in the “Messages” tab



# What are precision, recall and F1?

---

- Precision is the proportion of annotations the ML PR created that were correct
- Recall is the proportion of correct annotations that the ML PR created
- $P = \text{correct} / (\text{correct} + \text{spurious}) = \text{tp} / (\text{tp} + \text{fp})$
- $R = \text{correct} / (\text{correct} + \text{missing}) = \text{tp} / (\text{tp} + \text{fn})$
- 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
  - $F = 1 / ( \beta/P + (1-\beta)/R )$
  - $F1 = 2PR / (R + P)$
  - The equally balanced F1 ( $\beta = 0.5$ ) is the most common F-measure
- We can also run our own ML evaluation using the Corpus QA tool—let's do that now

# Splitting into training and test corpora



- 
- As mentioned earlier, to truly know how well a machine learner is performing, you need to test it on data that it was not trained on
  - We need separate test and training corpora
  - So now we are going to split our corpus in two

# Saving and splitting the corpus

ml-hands-on

Name	Size	Type
▶ corpus	93 items	folder
▶ test	0 items	folder
▶ training	4 items	folder
CreateMention.jape	571 bytes	plain text
ml-config-file.xml	1.8 KB	XML document

- Right click on your corpus and select “Save as XML”
- Create a new folder called “training” and save the documents into this folder
- Create a new directory alongside it called “test”
- In the file manager, cut half the documents out of “training” and paste them into “test” (try to randomise them a bit)

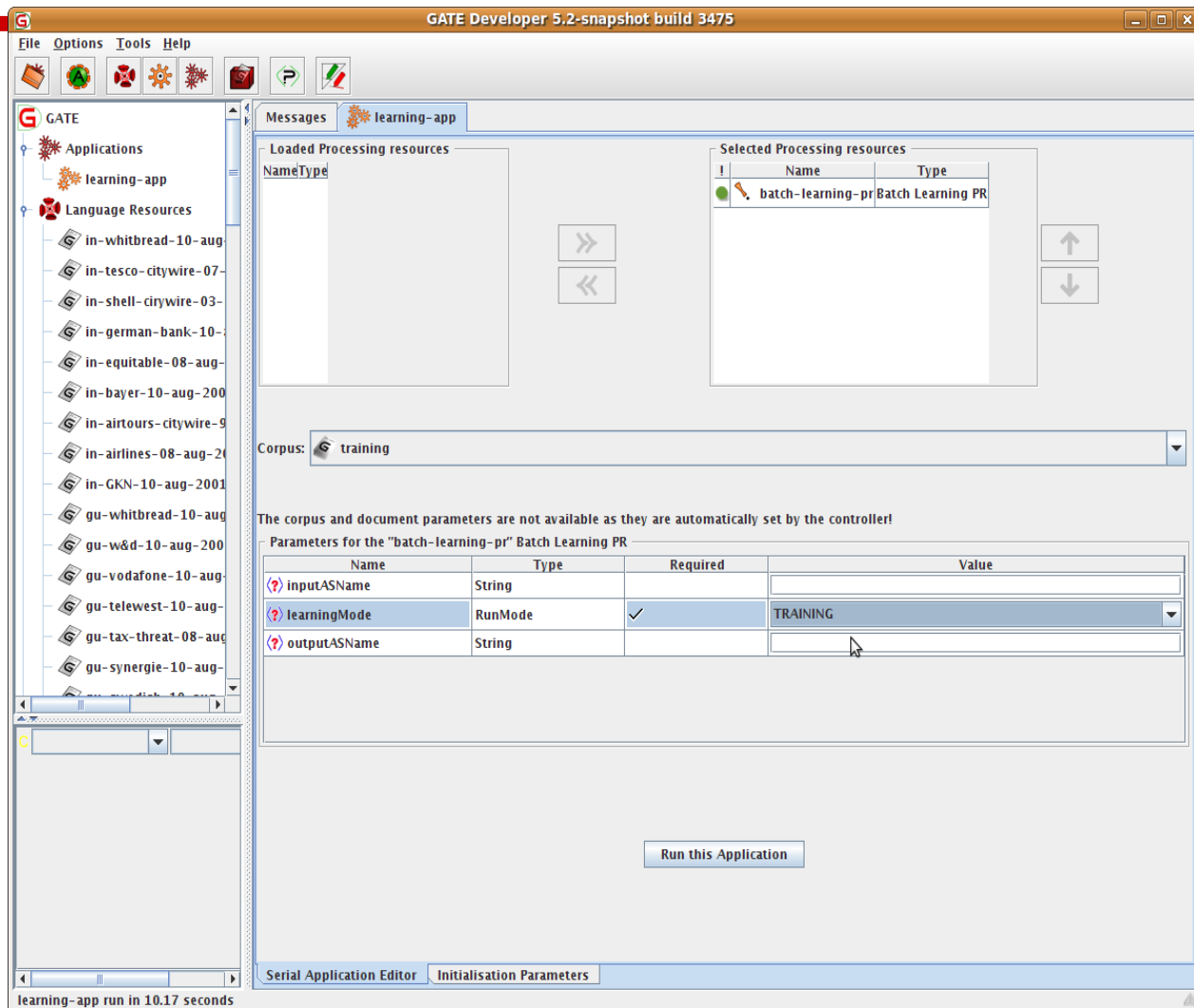


# Tidying up

---

- **Close all your open documents and processing resources in GATE Developer**
- **Close the modified ANNIE application recursively**
- **Create new corpora called “training” and “test”**
- **Populate your corpora with the documents you saved to disk**
  - **As before, use UTF-8**

# Running the ML PR in Training Mode



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

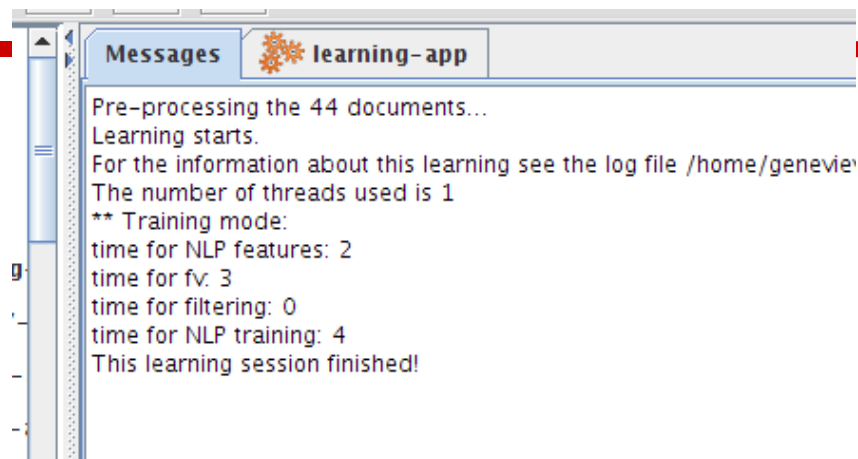
- Messages:** learning-app
- Loaded Processing resources:** (Empty)
- Selected Processing resources:**

Name	Type
batch-learning-pr	Batch Learning PR
- Corpus:** training
- Parameters for the "batch-learning-pr" Batch Learning PR:**

Name	Type	Required	Value
inputASName	String		
learningMode	RunMode	✓	TRAINING
outputASName	String		
- Buttons:** Run this Application
- Serial Application Editor:** Initialisation Parameters
- Status Bar:** learning-app run in 10.17 seconds

- Check that your PR is set to run on the training corpus
- Change the learningMode to "TRAINING" (the outputASName doesn't matter)
- Run the application

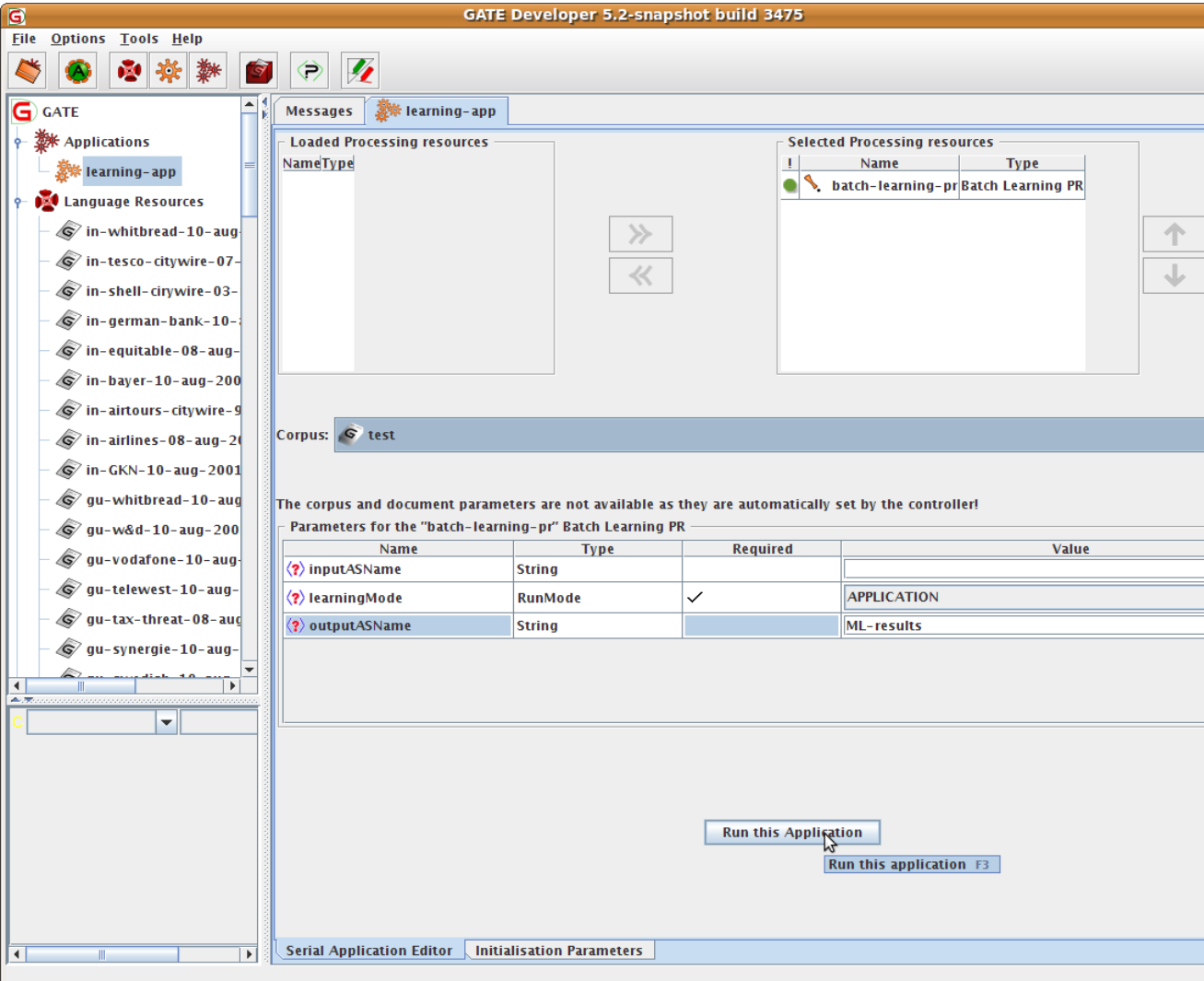
# Finished Training!

A screenshot of the GATE Messages window. The window title is "Messages" and the active tab is "learning-app". The message text is as follows:

```
Pre-processing the 44 documents...
Learning starts.
For the information about this learning see the log file /home/genevie
The number of threads used is 1
** Training mode:
time for NLP features: 2
time for fv: 3
time for filtering: 0
time for NLP training: 4
This learning session finished!
```

- Training may take a few minutes
- This time there is no evaluation result in the messages tab

# Running the ML PR in Application Mode

GATE Developer 5.2-snapshot build 3475

File Options Tools Help

GATE

Applications

- learning-app

Language Resources

- in-whitbread-10-aug
- in-tesco-citywire-07
- in-shell-citywire-03
- in-german-bank-10
- in-equitable-08-aug
- in-bayer-10-aug-200
- in-airtours-citywire-9
- in-airlines-08-aug-20
- in-GKN-10-aug-2001
- gu-whitbread-10-aug
- gu-w&d-10-aug-200
- gu-vodafone-10-aug
- gu-telewest-10-aug
- gu-tax-threat-08-aug
- gu-synergie-10-aug

Messages learning-app

Loaded Processing resources

NameType

Selected Processing resources

Name	Type
batch-learning-pr	Batch Learning PR

Corpus: test

The corpus and document parameters are not available as they are automatically set by the controller!

Parameters for the "batch-learning-pr" Batch Learning PR

Name	Type	Required	Value
inputASName	String		
learningMode	RunMode	✓	APPLICATION
outputASName	String		ML-results

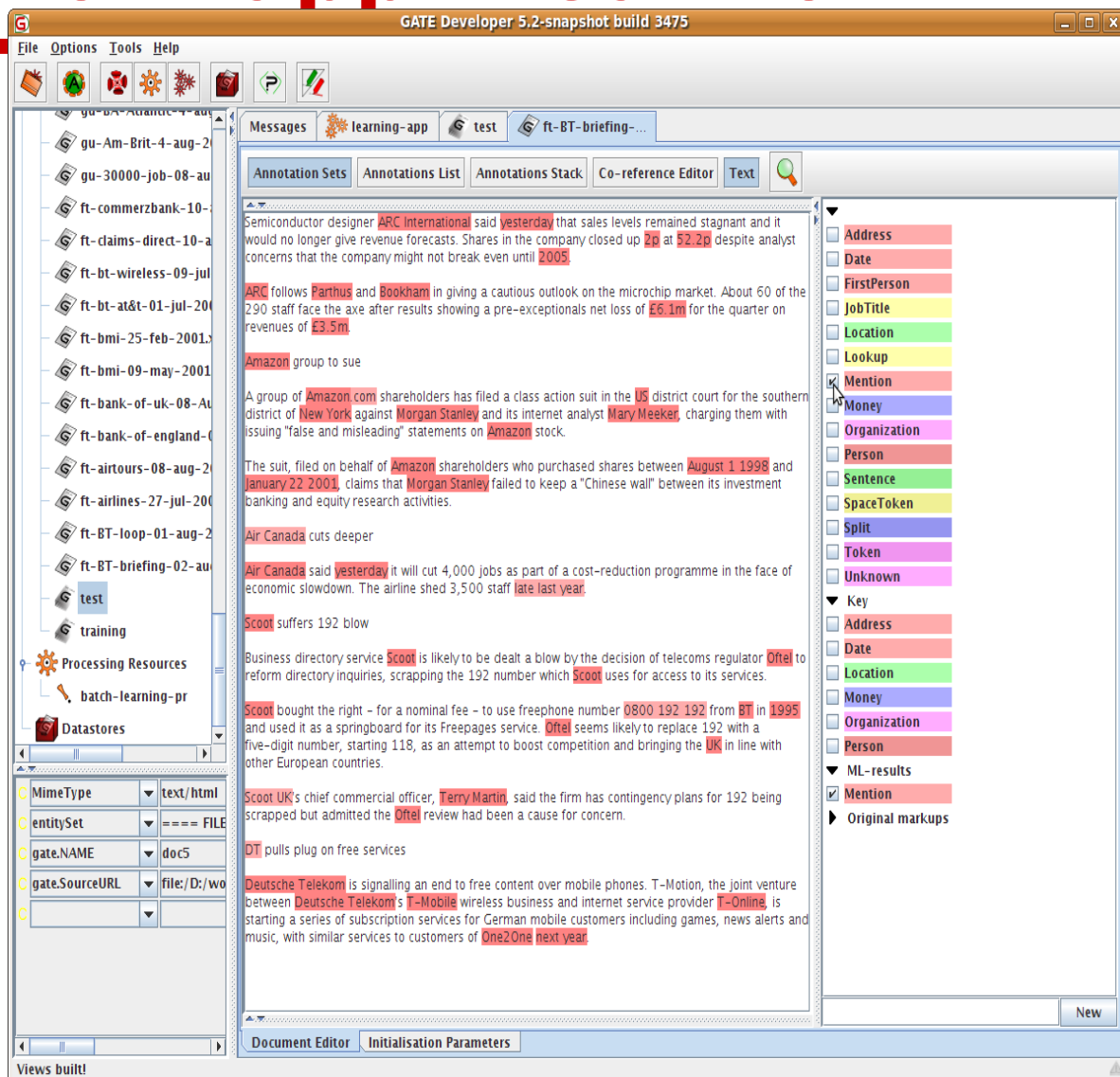
Run this Application

Run this application F3

Serial Application Editor Initialisation Parameters

- Change corpus to "test"
- Change learningMode to "APPLICATION"
- Set outputASName to "ML": your new annotations will go here, and you don't want to get them mixed up with the existing ones!

# Examining the results of application



The screenshot shows the GATE Developer interface. The main window displays a document with several paragraphs of text. Annotations are visible, including mentions of companies like ARC International, Amazon, Air Canada, and Scoot, and dates like yesterday, 2005, August 1 1998, and January 22 2001. The right-hand pane shows a list of annotation sets, with 'Mention' selected. The bottom pane shows the 'Document Editor' and 'Initialisation Parameters' tabs.

- Choose a document from the test corpus to look at
- You should have a new annotation set, created by the ML application
- There will be a “Mention” type both in the new set and the original
- They are similar but not identical!
- How similar do they appear to be? Do you think you will get a good result?

# Comparing the Sets with GATE

## Corpus QA

GATE Developer 5.2-snapshot build 3475

File Options Tools Help

Messages learning-app test ft-BT-briefing...

Annotation	Match	Only A	Only B	Overlap	Rec.B/A	Prec.B/A	F1-strict
Mention	1670	276	133	109	0.81	0.87	0.84
Macro summary					0.81	0.87	0.84
Micro summary	1670	276	133	109	0.81	0.87	0.84

Annotation Sets A & B

- [Default set] (A)
- Key
- ML-results (B)
- Original markups

present in every document

Annotation Types

- Lookup
- Mention
- Money
- Organization
- Percent

present in every selected set

Annotation Features

- class
- prob

present in every selected type

Measures

F-Score Classification

- F1-score strict
- F1-score lenient
- F1-score average

Compare

Compare annotations between sets A

Corpus editor Initialisation Parameters Corpus Quality Assurance

Views built!

- Select the test corpus and click on the Corpus QA tab (it will take a few seconds to scan the document)
- Select the Default and ML annotation sets
- Select the "Mention" type
- Select the "type" feature
- Choose an F-measure
- Click on Compare
- Did you get a good result? How does it compare to the result you got using evaluation mode?

# Using Annotation Diff to examine performance

Annotation Difference

Key doc: ft-BT-briefing-02-a... Key set: [Default set] Type: Mention Weight: 1.0

Resp. doc: ft-BT-briefing-02-a... Resp. set: ML-results Features:  all  some  none 1.0

Start	End	Key	Features	=?	Start	End	Response	Features
1517	1519	BT	{class=organization}	=	1517	1519	BT	{class=organization, prob=1.0}
171	173	2p	{class=money}	=	171	173	2p	{class=money, prob=1.0}
1956	1972	Deutsche · Telekom	{class=organization}	=	1956	1972	Deutsche · Telekom	{class=organization, prob=1.0}
46	55	yesterday	{class=date}	=	46	55	yesterday	{class=date, prob=1.0}
1322	1327	Oftel	{class=organization}	=	1322	1327	Oftel	{class=organization, prob=1.0}
867	882	January · 22 · 2001	{class=date}	=	867	882	January · 22 · 2001	{class=date, prob=1.0}
1198	1203	Scoot	{class=organization}	=	1198	1203	Scoot	{class=organization, prob=1.0}
514	524	Amazon.com	{class=organization}	~	514	520	Amazon	{class=organization, prob=1.0}
1753	1761	Scoot · UK	{class=organization}	-?				
1181	1195	late · last · year	{class=date}	-?				
1007	1017	Air · Canada	{class=organization}	-?				
1924	1926	DT	{class=organization}	-?				
				?-	1499	1511	0800 · 192 · 192	{class=money, prob=1.0}
482	488	Amazon	{class=organization}	<>	482	488	Amazon	{class=location, prob=0.99999946}
800	806	Amazon	{class=organization}	<>	800	806	Amazon	{class=location, prob=0.99999905}
756	762	Amazon	{class=organization}	<>	756	762	Amazon	{class=location, prob=1.0}

93 documents loaded

Correct: 36 Recall Precision F-measure

Partially correct: 1 Strict: 0.82 0.88 0.85

Missing: 7 Lenient: 0.84 0.90 0.87

False positives: 4 Average: 0.83 0.89 0.86

Statistics Adjudication

Show document

Export to HTML

- Switch to the “Document statistics” tab
- Choose a document
- Click on the Annotation Diff icon
- What kind of mistakes did your application make?



# Varying the configuration file

---

- Now we are going to experiment with varying the configuration file to see if we can produce varied results
- You can edit the configuration file in your text editor
- Make sure you save your changes then **reinitialise the PR!**



# Exercises

---

- **Spend some time working on your exercise sheet**
- **Feel free to ask questions**



# Confidence Thresholds

---

```
<PARAMETER name="thresholdProbabilityEntity" value="0.2"/>  
<PARAMETER name="thresholdProbabilityBoundary" value="0.42"/>  
<PARAMETER name="thresholdProbabilityClassification" value="0.5"/>
```

- Each classifier will provide confidence ratings—how likely is a result to be correct; we must determine how certain is good enough
- Depending on the application we might prefer to include or exclude annotations for which the learner is not too sure
- `thresholdProbabilityBoundary` and `thresholdProbabilityEntity` are thresholds for chunk learning
- `thresholdProbabilityClassification` applies to classification tasks, such as relation learning



# Classification tasks

---

- Example: the documents contains spans of text, which you want to classify as positive, negative, or neutral.
- This will be covered in more detail in Module 12 (Opinion Mining) tomorrow, with hands-on work



# Classification tasks

---

- `thresholdProbabilityClassification`: the “pickiness” of the classifiers
  - increasing this generally raises precision and reduces recall
  - decreasing this generally increases recall and reduces precision
- `thresholdProbabilityBoundary` and `thresholdProbabilityEntity`: ignored



# Classification tasks

---

- `<SURROUND VALUE="FALSE"/>`
- INSTANCE-TYPE: type of annotation that covers each span of text to classify
- Typically use NGRAM elements as attributes
- The GATE user guide gives examples

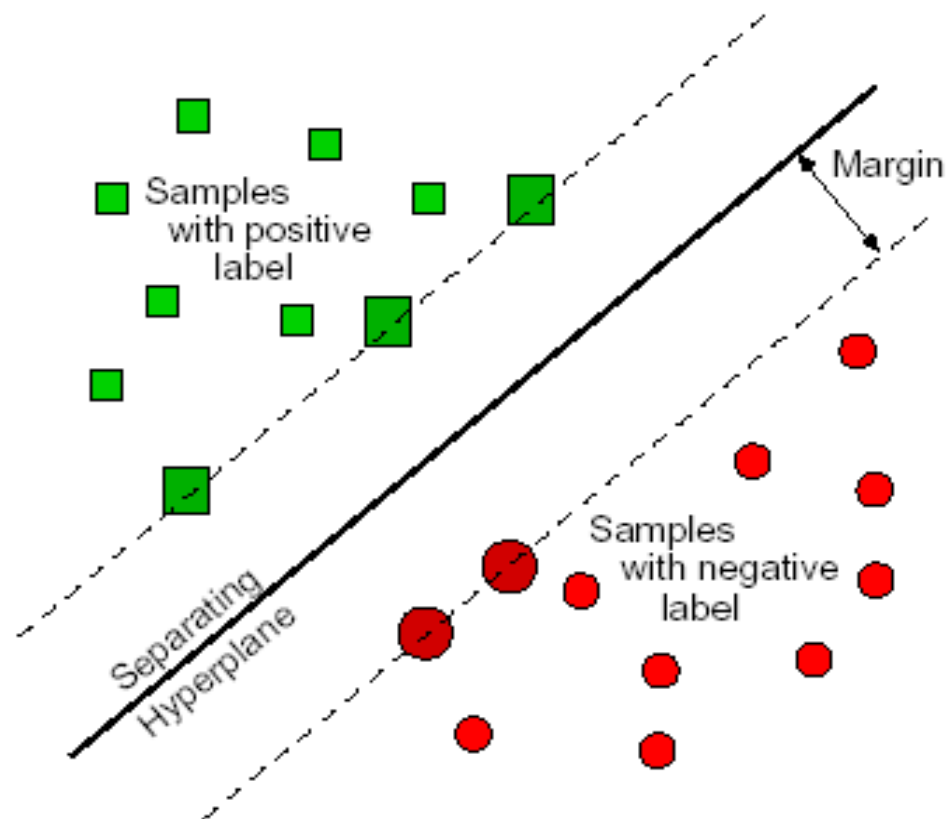


---

# Engines and Algorithms

# Support Vector Machines

- Attempt to find a hyperplane that separates data
- Goal: maximize margin separating two classes
- Wider margin = greater generalisation





# Support Vector Machines

---

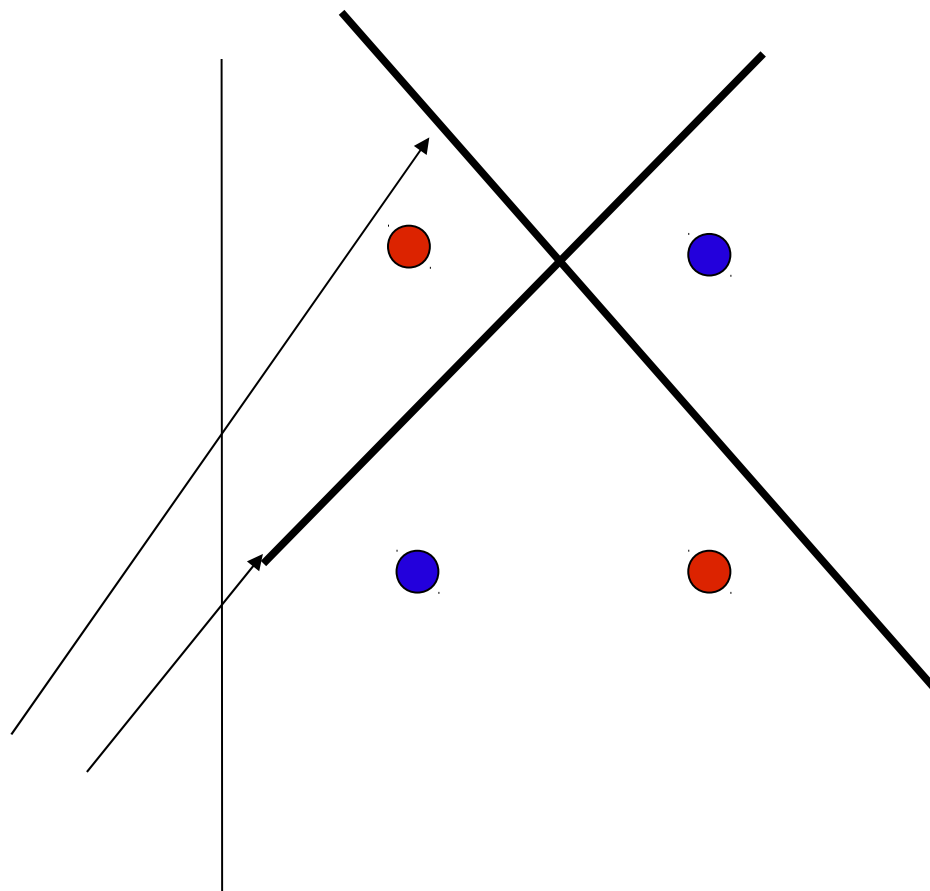
- Points near decision boundary: support vectors (removing them would change boundary)
- Points far from boundary not important for decision
- What if data doesn't split?
  - Soft boundary methods exist for imperfect solutions
  - However linear separator may be completely unsuitable

# Support Vector Machines

GATE

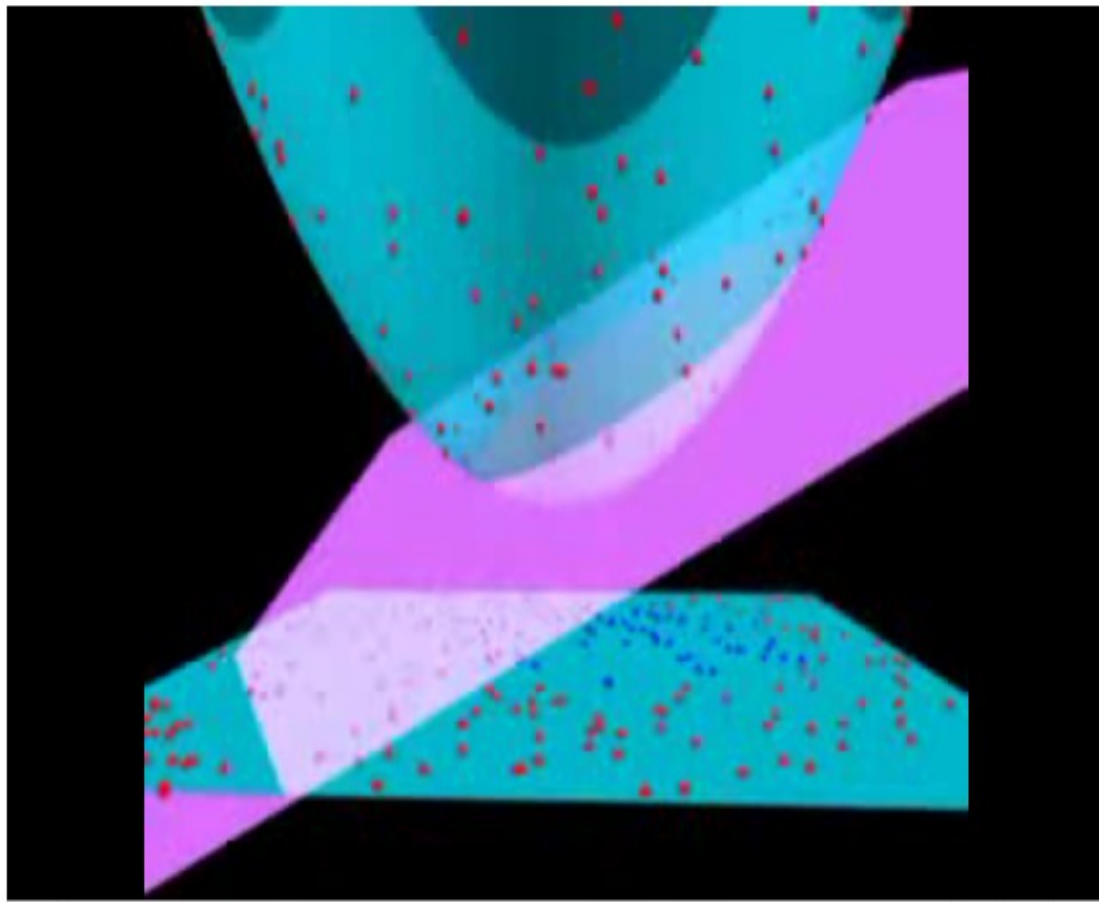
- What if there is no separating hyperplane?
- See example:
- Or class may be a globule

They do not work!



# Kernel Trick

- Map data into different dimensionality
- <http://www.youtube.com>
- 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 ~~GATE~~ NLP

---

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



# Support Vector Machines

---

- SVMs combined with kernel trick provide a powerful technique
- Multiclass methods simple extension to two class technique (one vs. another, one vs. others)
- Widely used with great success across a range of linguistic tasks

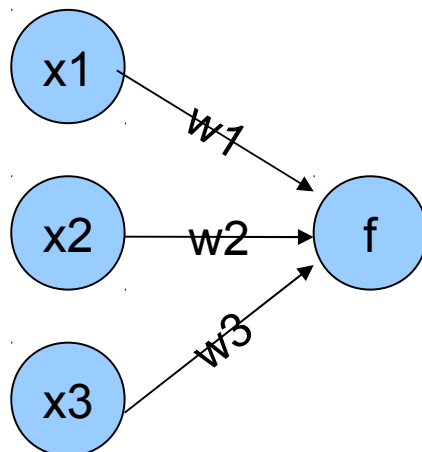


# Perceptron and PAUM

---

- Perceptron is one of the oldest ML methods (invented in the 50s!)
- Has some similarities to SVM (it determines a hyperplane separator)
- Theoretically SVM works a little better because it calculates the optimal separator
- In practice, however, there is usually little difference, and Perceptron is a lot faster!

# Perceptron



- You might think of perceptrons as being these things (correct)
- What this is actually calculating is a dot product  $w \cdot x$

# More perceptron

---

$$f(x) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

- $x$  is a datapoint represented as a vector
- $w$  is a vector that defines the separating hyperplane (it is perpendicular to it)
- This function tells you which side of the hyperplane your point lies
- $b$  defines an offset from the origin

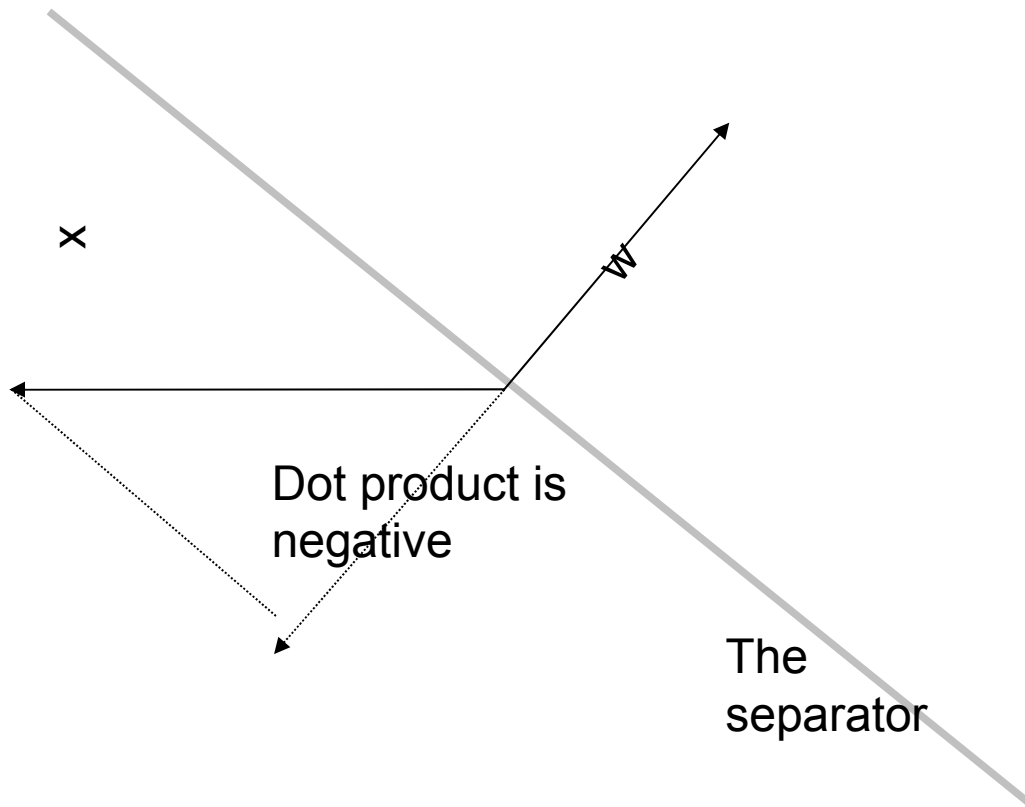


# More perceptron

---

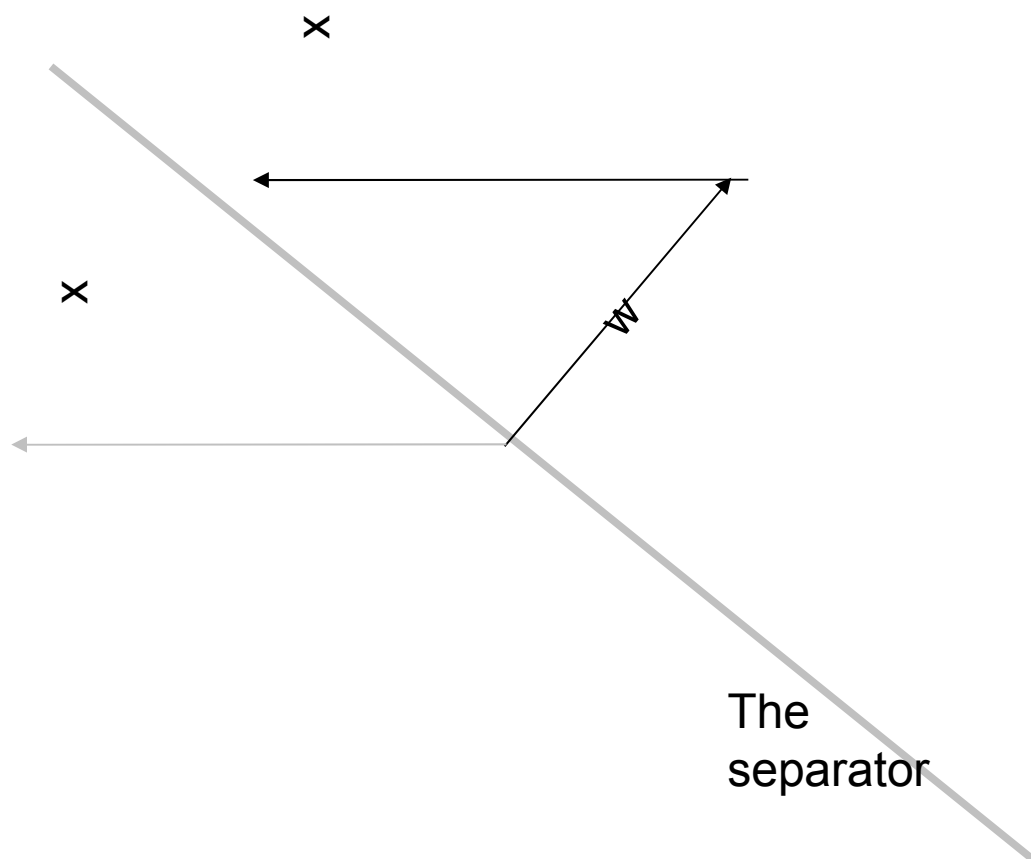
- How does it learn?
  - Each datapoint is annotated with class value 1 or 0
  - Function returns 1 or 0 depending on which side of the separator the point lies
  - Calculate difference between actual and desired output
  - Multiply input vector by this delta and add it to the weight vector
  - Given sufficient iterations the separator will find a solution

# Perceptron update



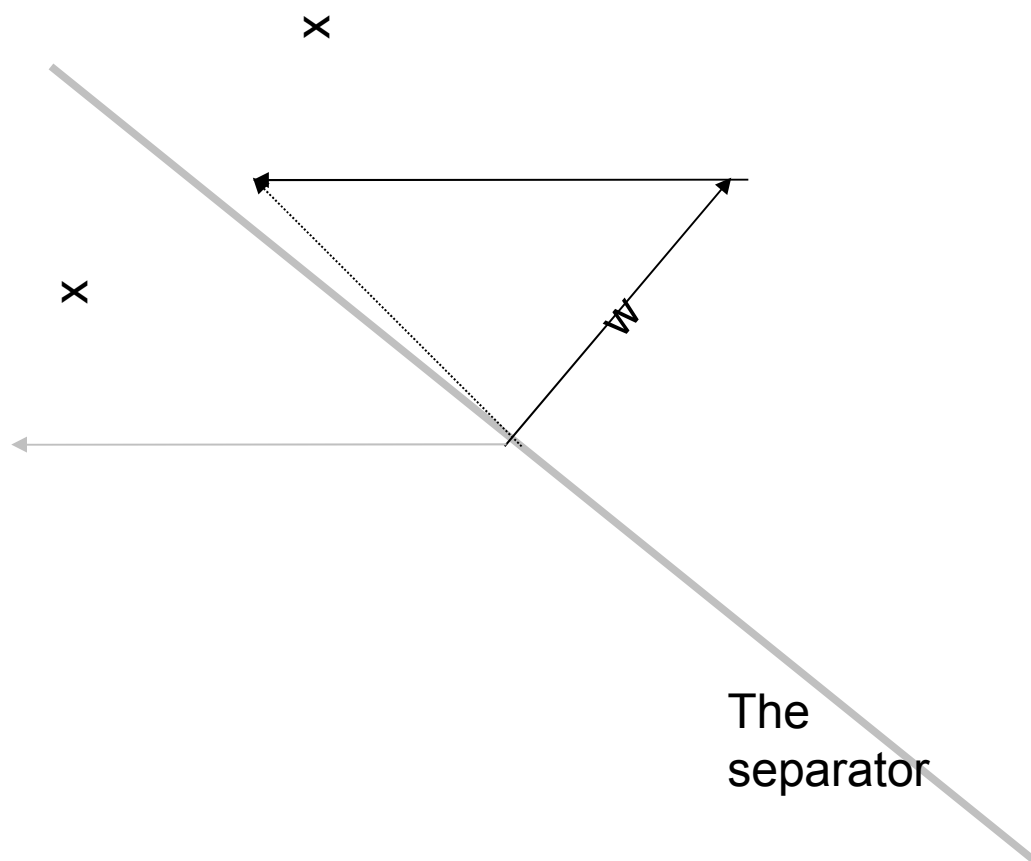
- Dot product is negative, so  $f=0$
- But  $x$  is a positive example!
- Oh no! Must update

# Perceptron update



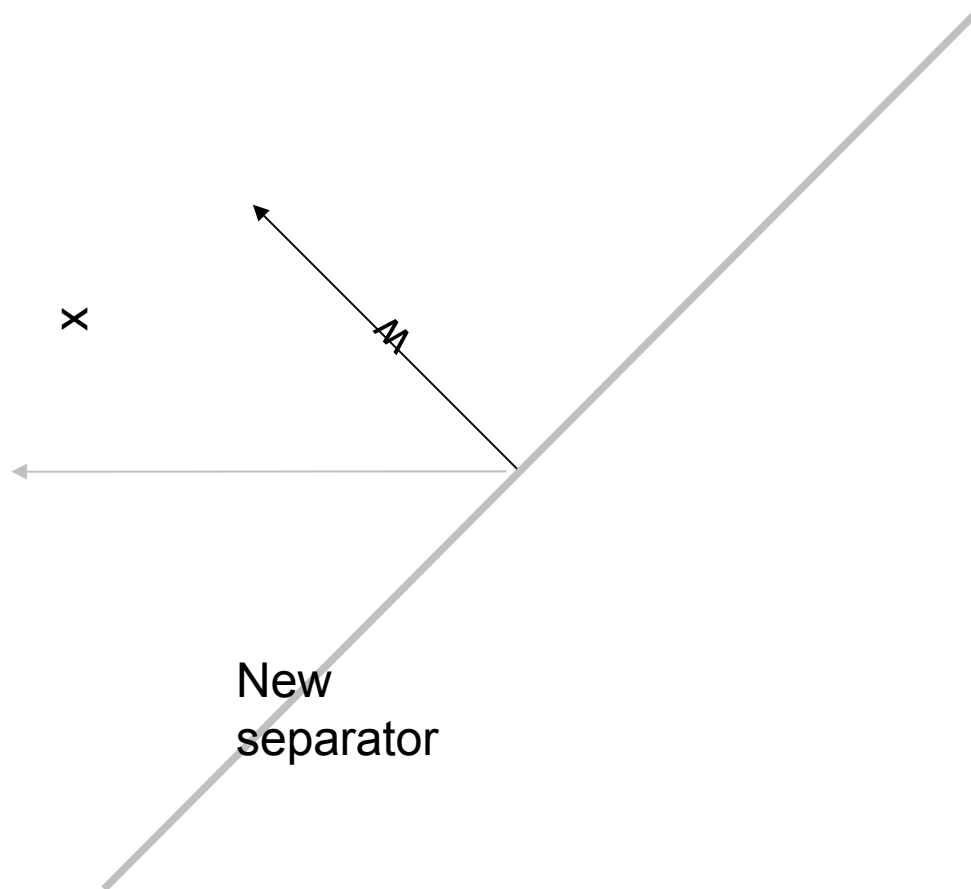
- $x$  class is 1
- $f(x) = 0$
- $w += (1-0)x$

# Perceptron update



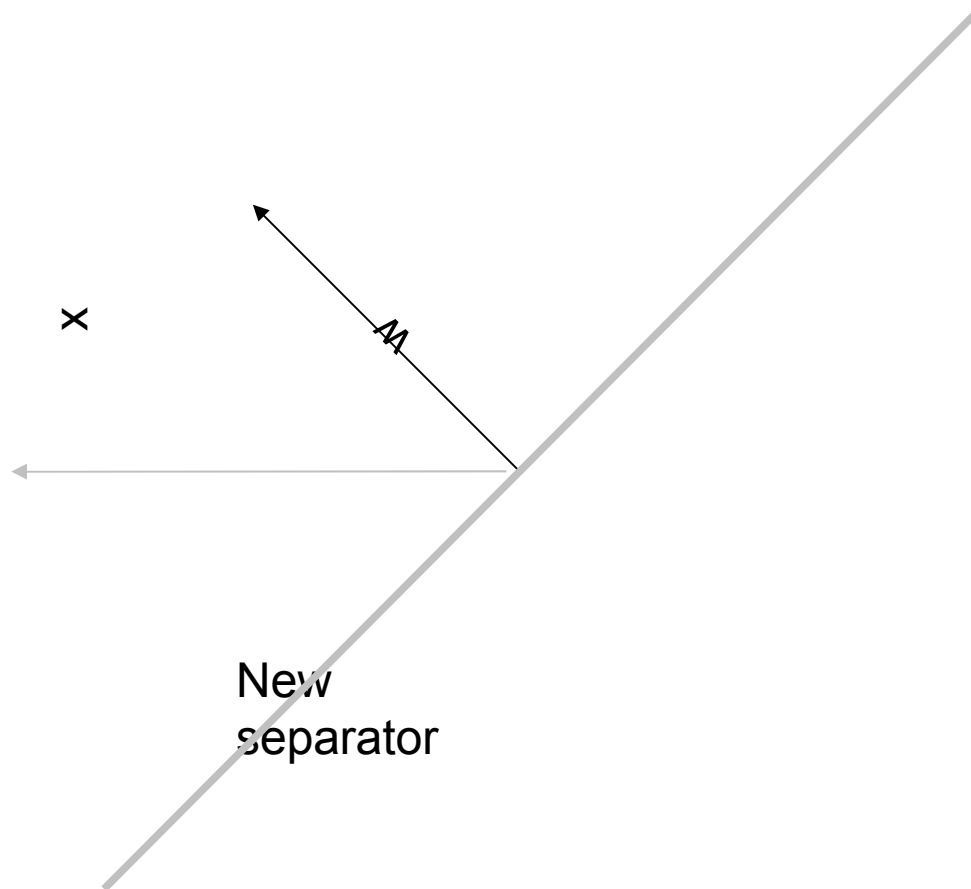
- $x$  class is 1
- $f(x) = 0$
- $w += (1-0)x$

# Perceptron update



- $x$  class is 1
- $f(x) = 0$
- $w += (1-0)x$

# Perceptron update



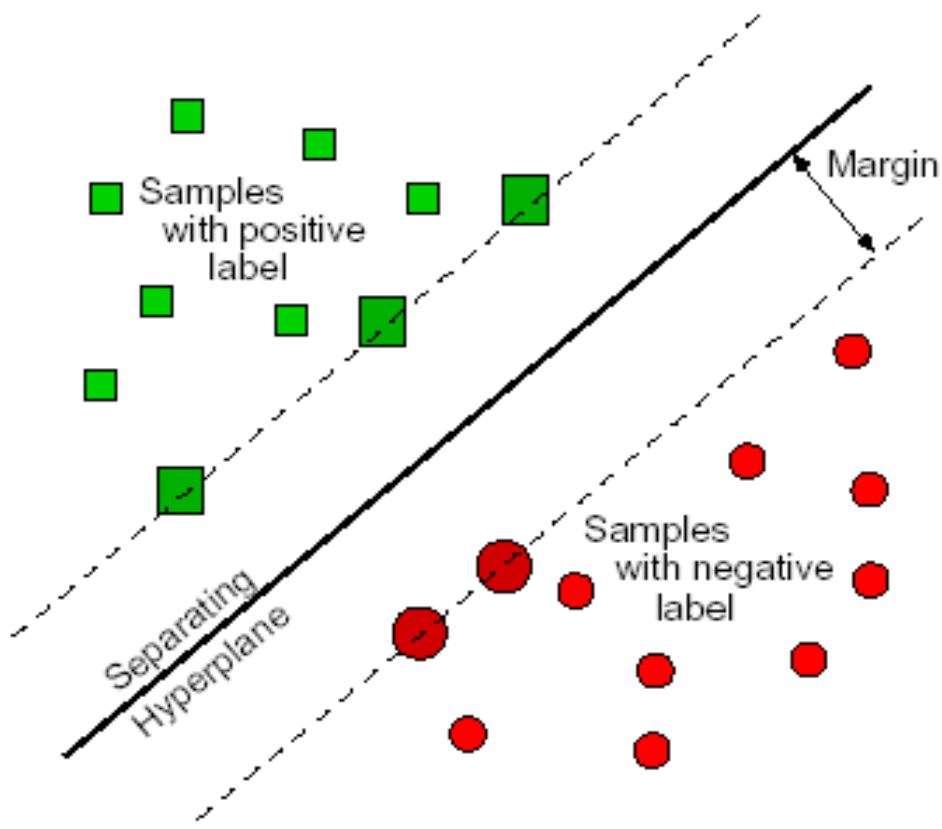
- Now  $x$  is on the right side of the separator!

# Perceptron with Uneven Margins

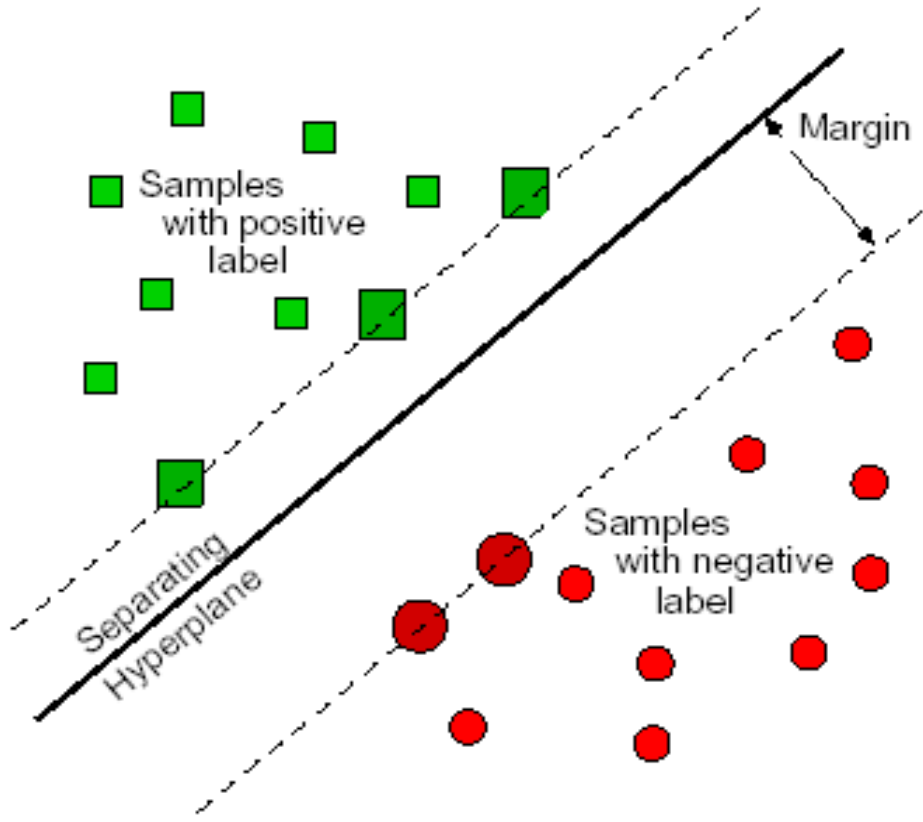


- Both Perceptron and SVM implement “uneven margins”
- (PAUM stands for Perceptron Algorithm with Uneven Margins)
- This means that it doesn't position the separator centred between the points, but more towards one side

# Even Margins



# Uneven Margins





# Why Uneven Margins?

---

- In NLP the datasets are often very imbalanced
- For example if you are finding instances of “Person”, you will have very many words that are not people and only a few that are
- Uneven margins may help with this
- Y. Li, K. Bontcheva, and H. Cunningham. Using Uneven Margins SVM and Perceptron for Information Extraction. Proceedings of Ninth Conference on Computational Natural Language Learning (CoNLL-2005), pp. 72-79. 2005.



# Some Other Algorithms

---

- Batch Learning PR also includes the following from Weka
  - Naïve Bayes
    - Uses Bayes' theorem (probabilities) to determine the most likely class given attributes and training corpus
  - K-Nearest Neighbour
    - Determines class of a point based on k training points positioned geometrically closest to it
  - C4.5 (decision tree)
    - Makes a series of binary decisions that determine the class of a point based on its attribute values (e.g. “is string length > 3?”)