



Semantic CMS Community

Olivier Grisel
Nuxeo

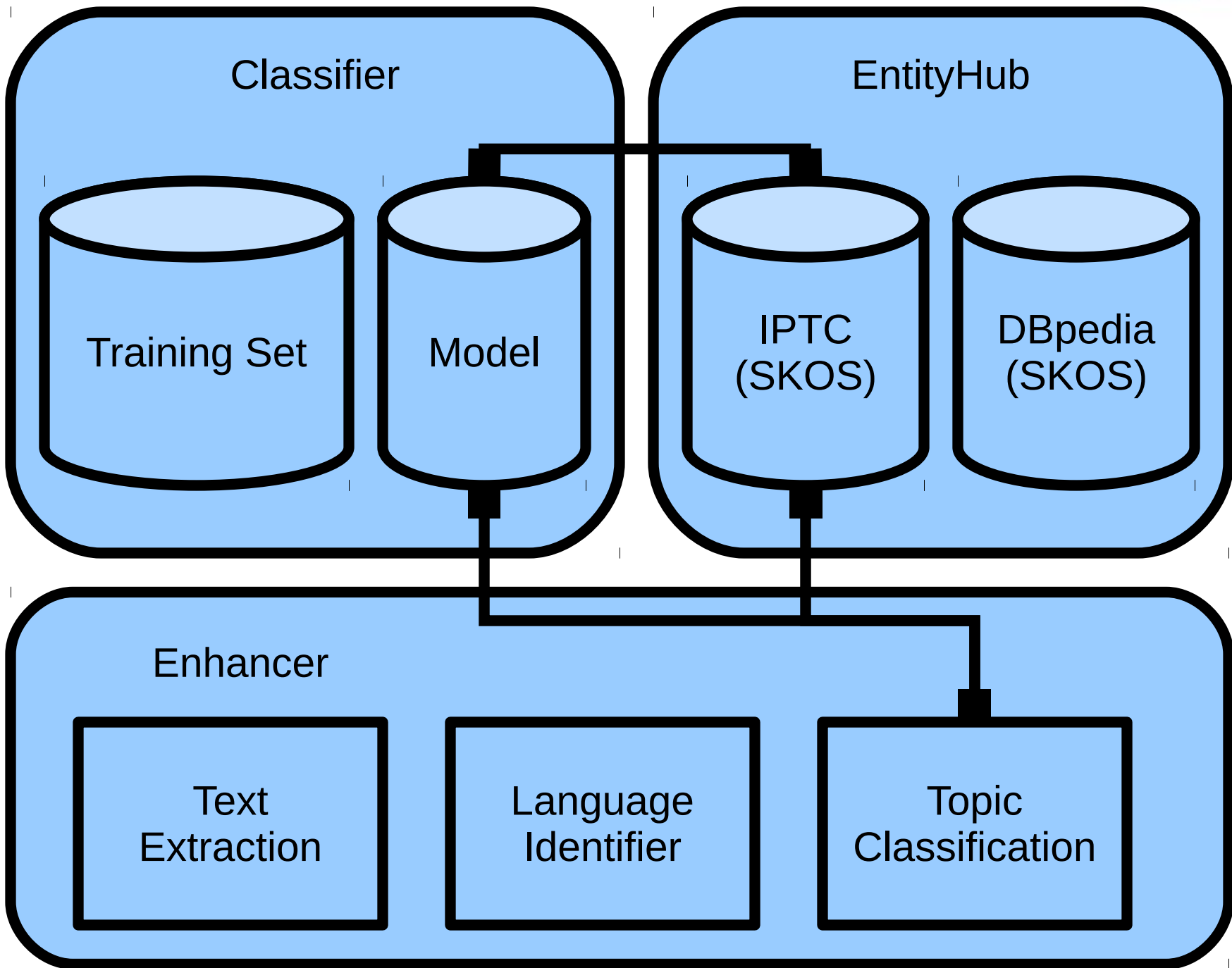
March, 2012



Co-funded by the
European Union

Topic Classification

The missing
documentation ;)



Building & Deploying

```
cd ~/stanbol/launchers/stable/target/  
java -jar org.apache.stanbol.launchers.stable-  
0.10.0-incubating-SNAPSHOT.jar -p 9090
```

```
cd ~/stanbol/enhancer/engines/topic  
mvn install -DskipTests -PinstallBundle \  
-Dsling.url=http://localhost:9090/system/console
```

```
cd ~/stanbol/enhancer/topic-web  
mvn install -DskipTests -PinstallBundle \  
-Dsling.url=http://localhost:9090/system/console
```

<http://localhost:9090/system/console/configMgr>

Create a new **TopicClassification** Engine configuration
Add values for **Name** and **Training Set** and hit “Save”

org.apache.stanbol.enhancer.engines.entitytagging.impl.NamedEntityTaggingEngine.5ceaa95d-ba72-412a-9bbc-c8d4bc98c974 Apache Stanbol Enhancer Enhancement Engine : Entity Tagging

Apache Stanbol Enhancer Engine: Topic Classification

Allows to categorize parsed content along topics.

Name	<input type="text" value="iptc-classifier"/> The name of the enhancement engine as used in the RESTful interface '/engine/<name>' (stanbol.enhancer.engine.name)
Engine Order	<input type="text" value="100"/> The order is used by the Weighted Chain to calculate the execution order of Enhancement Engines. Engines with lower values are executed before engines with higher values. Engines with the same value can be executed in parallel. (org.apache.stanbol.enhancer.engine.topic.order)
Solr Core	<input type="text"/> ----- (org.apache.stanbol.enhancer.engine.topic.positiveSupportField)
Negative Support Field	<input type="text" value="negative_support"/> org.apache.stanbol.enhancer.engine.topic.negativeSupportField.description (org.apache.stanbol.enhancer.engine.topic.negativeSupportField)
org.apache.stanbol.enhancer.engine.topic.trainingSetId.name	<input type="text" value="iptc-trainingset"/> org.apache.stanbol.enhancer.engine.topic.trainingSetId.description (org.apache.stanbol.enhancer.engine.topic.trainingSetId)
Ranking	<input type="text" value="0"/> If two enhancement engines with the same name are active the one with the higher ranking will be used to process parsed content items. (service.ranking)

Configuration Information

Persistent Identity (PID)	[Temporary PID replaced by real PID upon save]
Factory Persistent Identifier (Factory PID)	org.apache.stanbol.enhancer.engine.topic.TopicClassificationEngine
Configuration Binding	Unbound or new configuration

Save Reset Abort

<http://localhost:9090/system/console/configMgr>

Create a new Topic **Solr Training Set** configuration
Add value for **Name** and hit “Save”

The screenshot shows a configuration window titled "org.apache.stanbol.enhancer.topic.training.SolrTrainingSet.name". It contains several input fields for configuration parameters, each with a label and a description. The fields are:

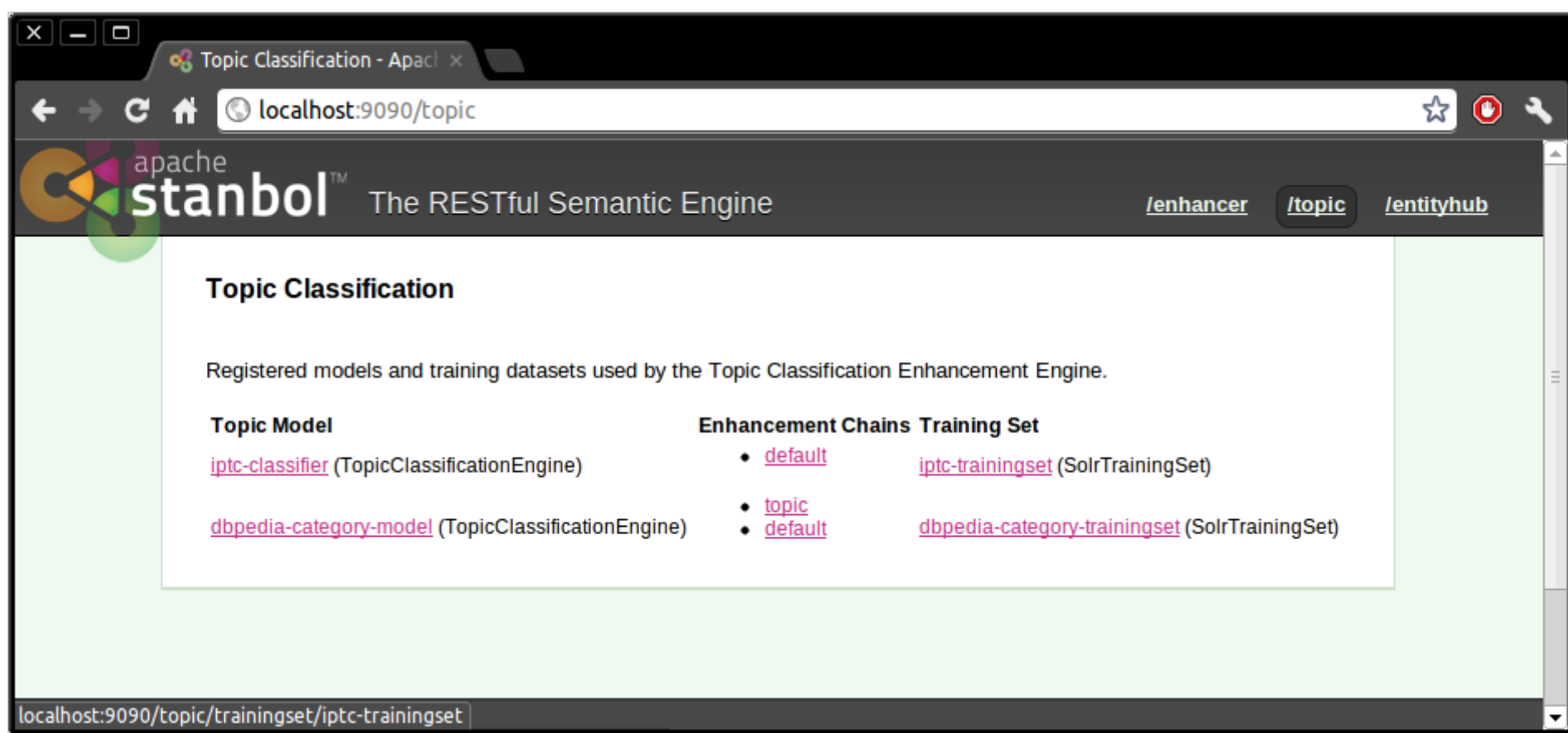
- org.apache.stanbol.enhancer.topic.training.SolrTrainingSet.description**: (empty)
- org.apache.stanbol.enhancer.topic.trainingset.id.name**: iptc-trainingset
- org.apache.stanbol.enhancer.topic.trainingset.id.description**: (org.apache.stanbol.enhancer.topic.trainingset.id)
- Solr Core**: (empty)
- org.apache.stanbol.enhancer.engine.topic.solrCore.description**: (org.apache.stanbol.enhancer.engine.topic.solrCore)
- org.apache.stanbol.enhancer.engine.topic.exampleIdField.name**: id
- org.apache.stanbol.enhancer.engine.topic.exampleIdField.description**: (org.apache.stanbol.enhancer.engine.topic.exampleIdField)
- org.apache.stanbol.enhancer.engine.topic.exampleTextField.name**: text
- org.apache.stanbol.enhancer.engine.topic.exampleTextField.description**: (org.apache.stanbol.enhancer.engine.topic.exampleTextField)
- org.apache.stanbol.enhancer.engine.topic.topicsUriField.name**: topics
- org.apache.stanbol.enhancer.engine.topic.topicsUriField.description**: (org.apache.stanbol.enhancer.engine.topic.topicsUriField)
- org.apache.stanbol.enhancer.engine.topic.modificationDateField.name**: modification_dt
- org.apache.stanbol.enhancer.engine.topic.modificationDateField.description**: (org.apache.stanbol.enhancer.engine.topic.modificationDateField)

Below the input fields is a "Configuration Information" section with the following details:

Persistent Identity (PID)	org.apache.stanbol.enhancer.topic.training.SolrTrainingSet.d32de556-ce53-42f9-9a10-06c11a22649d
Factory Persistent Identifier (Factory PID)	org.apache.stanbol.enhancer.topic.training.SolrTrainingSet
Configuration Binding	Apache Stanbol Enhancer Enhancement Engine : Topic Classification (org.apache.stanbol.enhancer.engine.topic), Version 0.10.0.incubating-SNAPSHOT

At the bottom right, there are three buttons: "Save", "Reset", and "Abort".

http://localhost:9090/topic



The screenshot shows a web browser window with the Apache Stanbol RESTful Semantic Engine interface. The browser's address bar displays "localhost:9090/topic". The page header includes the Apache Stanbol logo and the text "The RESTful Semantic Engine". Navigation links for "/enhancer", "/topic" (which is highlighted), and "/entityhub" are visible. The main content area is titled "Topic Classification" and contains the following text: "Registered models and training datasets used by the Topic Classification Enhancement Engine." Below this, there are two columns of information: "Topic Model" and "Enhancement Chains Training Set".

Topic Model	Enhancement Chains	Training Set
iptc-classifier (TopicClassificationEngine)	<ul style="list-style-type: none">default	iptc-trainingset (SolrTrainingSet)
dbpedia-category-model (TopicClassificationEngine)	<ul style="list-style-type: none">topicdefault	dbpedia-category-trainingset (SolrTrainingSet)

The browser's address bar at the bottom shows the path "localhost:9090/topic/trainingset/iptc-trainingset".

Registering concepts

```
$ curl -X POST http://localhost:8080/topic/model\  
/iptc-classifier/concept?id=concept_1
```

```
$ curl -X POST http://localhost:8080/topic/model\  
/iptc-classifier/concept?id=concept_2
```

```
$ curl -X POST http://localhost:8080/topic/model\  
/iptc-classifier/concept?id=concept_3\  
&broadener=concept_1&broadener=concept_2
```

Registering concepts from SKOS

```
$ curl -X POST --data @iptc-skos.rdf.xml  
http://localhost:8080/topic/model/iptc-classifier
```


Loading a Training Set

```
$ curl -X POST --data @file_1.txt \  
http://localhost:8080/topic/model\  
/iptc-classifier/trainingset\  
?example_id=example_1\  
&concept=concept_3&concept=concept_42
```

Importing a NewsML archive

```
$ cd ~/stanbol/enhancer/topic-web/tools
```

```
$ python newsmlexporter.py /path/to/newsm1/ 1000 \  
http://localhost:9090/topic/model/  
iptc-classifier/trainingset
```

```
Processed news 100/1000 in 01.855s  
Processed news 200/1000 in 01.893s  
Processed news 300/1000 in 02.012s  
Processed news 400/1000 in 01.771s  
Processed news 500/1000 in 01.919s  
Processed news 600/1000 in 01.823s  
Processed news 700/1000 in 01.876s  
Processed news 800/1000 in 01.793s  
Processed news 900/1000 in 01.875s  
Processed news 1000/1000 in 01.817s
```

Training the model

```
$ curl -X POST http://localhost:9090/topic/model/\
  iptc-classifier/trainer?incremental=true
Successfully updated the statistical model(s) of
200 concept(s).
```

```
$ curl -X POST http://localhost:9090/topic/model/\
  iptc-classifier/trainer?incremental=true
Successfully updated the statistical model(s) of 0
concept(s).
```

Future work

- Make it possible to plug alternative trainable classifier
 - The present Solr-based Approximate Rocchio Classifier that scales to large numbers of classes
 - OpenNLP Text Categorizer: Averaged Perceptron
 - Mahout's SGD Logistic Regression with the hashing trick for scaling to large number of classes