



De La Salle University

**FILIET: An Information Extraction System
For Filipino Disaster-Related Tweets**

A Technical Manual
Presented to
the Faculty of the College of Computer Studies
De La Salle University – Manila

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science

by
DELA CRUZ, Kyle Mc Hale B.
GARCIA, John Paul F.
KALAW, Kristine Ma. Dominique F.
LU, Vilson E.

REGALADO, Ralph Vincent
Adviser

April 27, 2015



Table of Contents

1.0	Introduction.....	1-1
2.0	System Module Class Diagram	2-1
2.1	Models.....	2-1
2.1.1	Sentence	2-1
2.1.2	Rules	2-2
2.1.3	Extracted Information	2-2
2.2	Crawler Module	2-3
2.3	Information Extraction Engine	2-4
2.3.1	Preprocessor Manager.....	2-5
2.3.2	Feature Extraction Module	2-8
2.3.3	Classifier Module.....	2-9
2.3.4	Rule Inductor Module	2-10
2.4	Ontology	2-11
2.4.1	OntologyModule	2-11
2.4.2	OntologyRetriever	2-12
2.4.3	Binder	2-12
2.5	Others.....	2-13
2.5.1	Database	2-13
2.5.2	Reader.....	2-13
2.5.3	Language Modeller	2-14
2.5.4	XML Parser	2-15
2.6	Entity Relationship Diagram.....	2-16
3.0	Class Dictionary.....	3-1
3.1	Models.....	3-2
3.1.1	Tweet Class.....	3-2
3.1.2	Sentence Class	3-4
3.1.3	Token Class	3-6
3.1.4	Rule Class	3-7
3.1.5	Grammar Class	3-8
3.1.6	ExtractedInformation Class	3-8
3.1.7	PostInformationExtracted Class.....	3-9
3.2	Crawler Module	3-10
3.2.1	Crawler Class	3-10
3.3	Information Extraction Engine	3-10
3.3.1	InformationEngineExtraction Class.....	3-10
3.4	Preprocessing Module	3-11
3.4.1	Preprocessor Manager Class.....	3-11
3.4.2	Normalizer	3-12
3.4.3	Tokenizer.....	3-13
3.4.4	POS Tagger	3-15
3.4.5	NER.....	3-16
3.5	Feature Extractor Module.....	3-18
3.5.1	FeatureExtraction Class	3-18
3.6	Classifier Module.....	3-19
3.6.1	ClassifierInterface Class	3-19
3.6.2	ClassifierBuilder Class	3-19
3.6.3	Classifier Class.....	3-20



3.6.4	ClassifierImpl Class.....	3-21
3.6.5	MultiClassifierImpl Class	3-21
3.7	Rule Induction Module	3-23
3.8	Ontology Module	3-23
3.8.1	OntologyModule Class	3-23
3.8.2	Tweet Class (Ontology).....	3-27
3.8.3	CallForHelpTweet Class	3-28
3.8.4	CasualtiesAndDamageTweet Class	3-28
3.8.5	CautionAndAdviceTweet Class	3-29
3.8.6	DonationTweet Class	3-30
3.8.7	OntologyRetriever Class	3-30
3.8.8	DPVPair Class.....	3-32
3.8.9	LabelledDPVPair Class.....	3-32
3.8.10	RetrievedTweet Class	3-33
3.8.11	Binder Class	3-34
3.9	Others.....	3-35
3.9.1	DBFactory Class	3-35
3.9.2	DBConnection Class	3-36
3.9.3	Reader Class.....	3-36
3.9.4	XmlParser Class	3-37
3.9.5	DocumentFrequency Class.....	3-37
3.9.6	WeightScorer Class	3-38
3.9.7	NGramModeller Class.....	3-39
3.9.8	Filter Class	3-40



1.0 Introduction

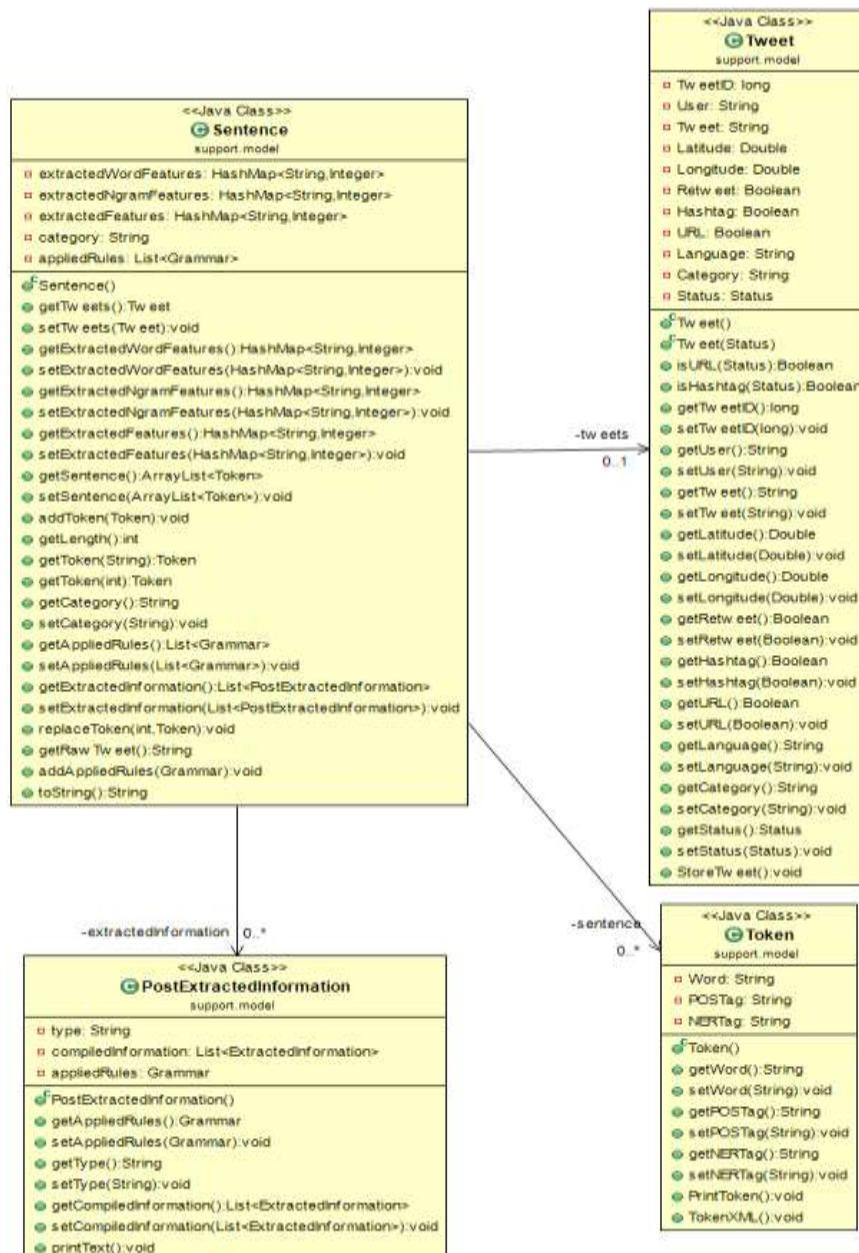
FILIET (Filipino Information Extraction for Twitter) is an information extraction system that makes use of handcrafted rules in order to extract the information from tweets composed in the Filipino language. The system is composed of six modules: the crawler, preprocessor, feature extraction, classification, rule inductor, and ontology module. The crawler module can be run as a standalone feature of the system whereas the rest are integrated. Through the crawler module, tweets are collected and stored in the database which is then exported to a CSV file. The remainder of the FILIET system makes use of the exported CSV file for extraction.



2.0 System Module Class Diagram

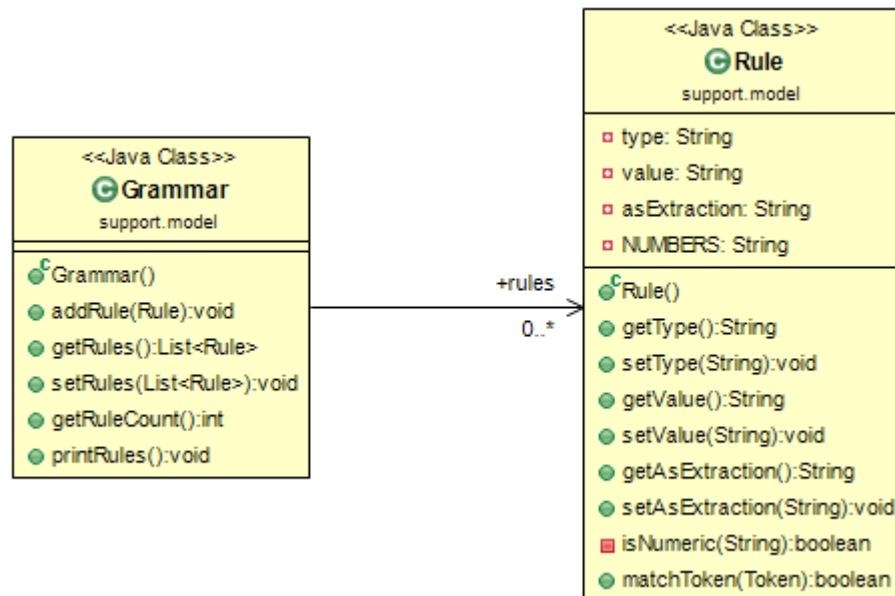
2.1 Models

2.1.1 Sentence

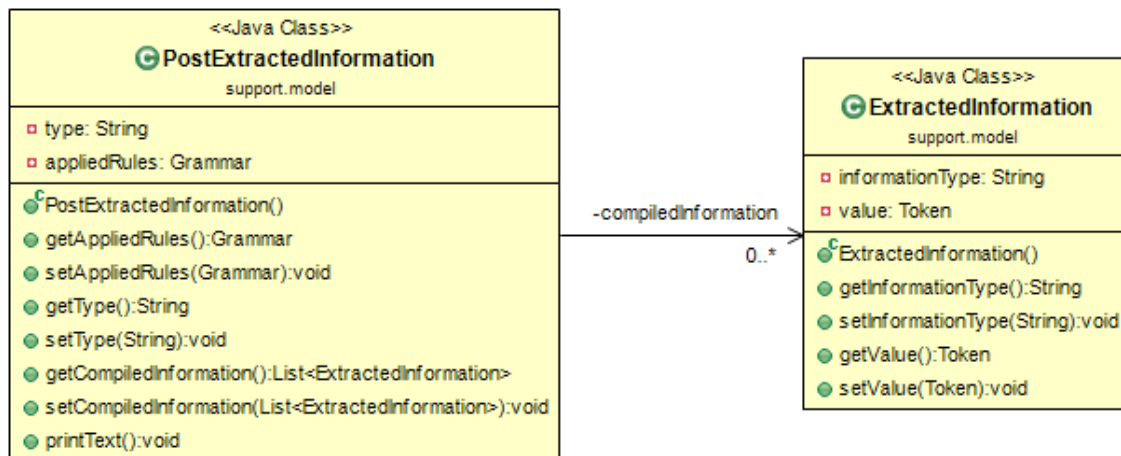




2.1.2 Rules

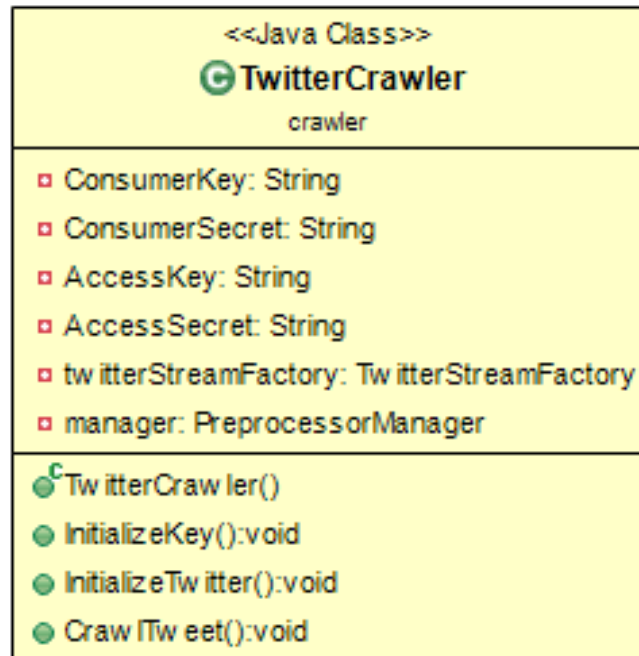


2.1.3 Extracted Information



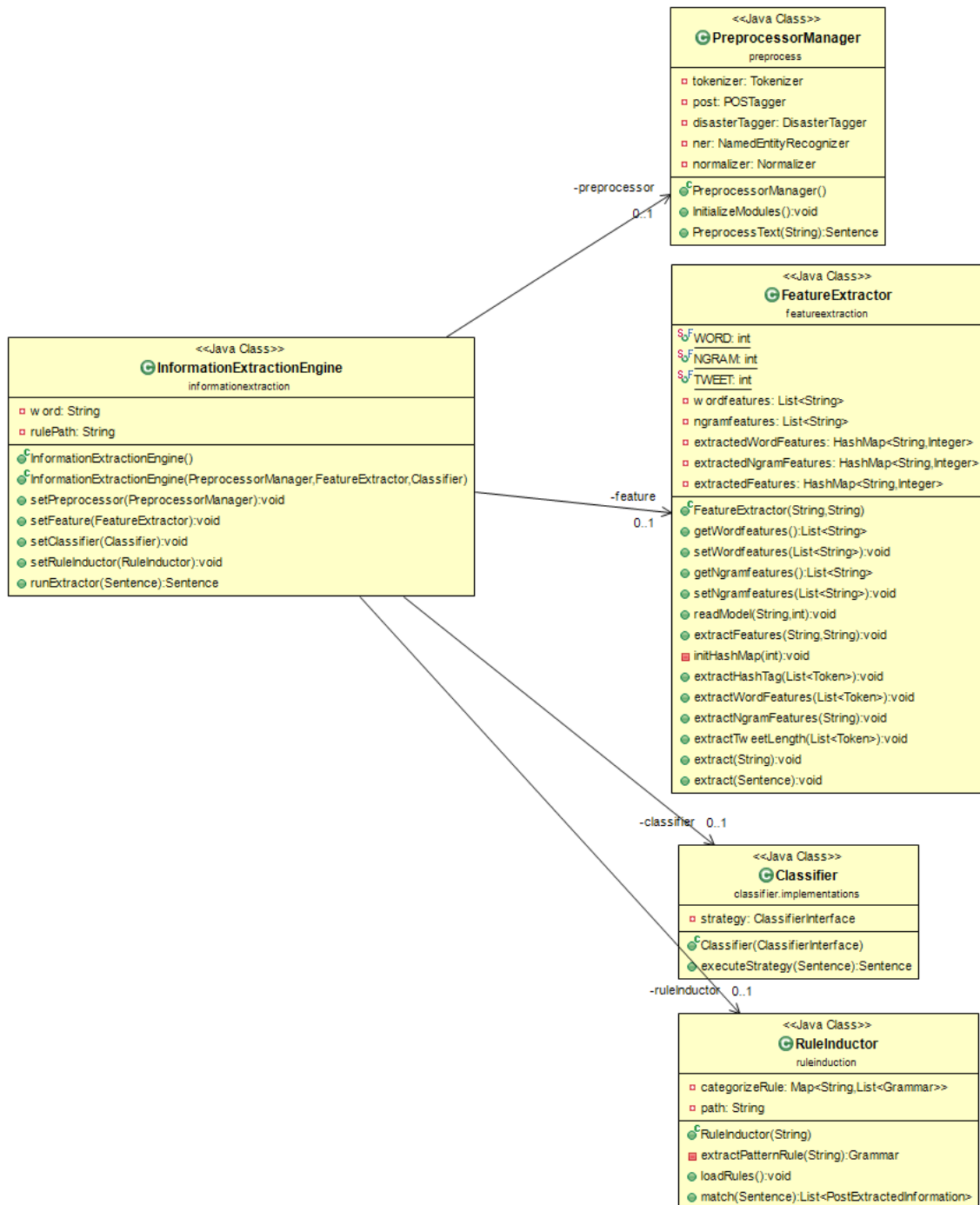


2.2 Crawler Module



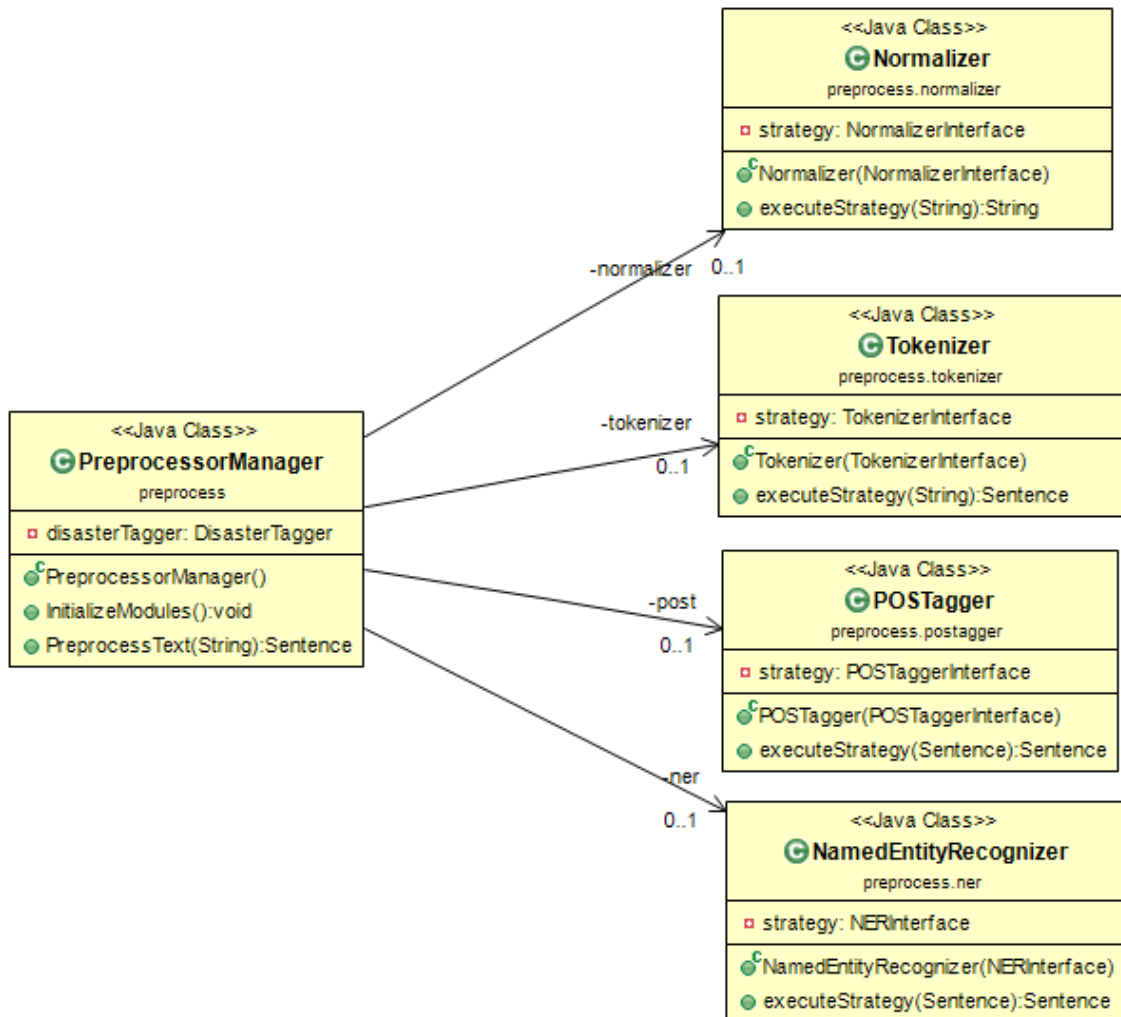


2.3 Information Extraction Engine



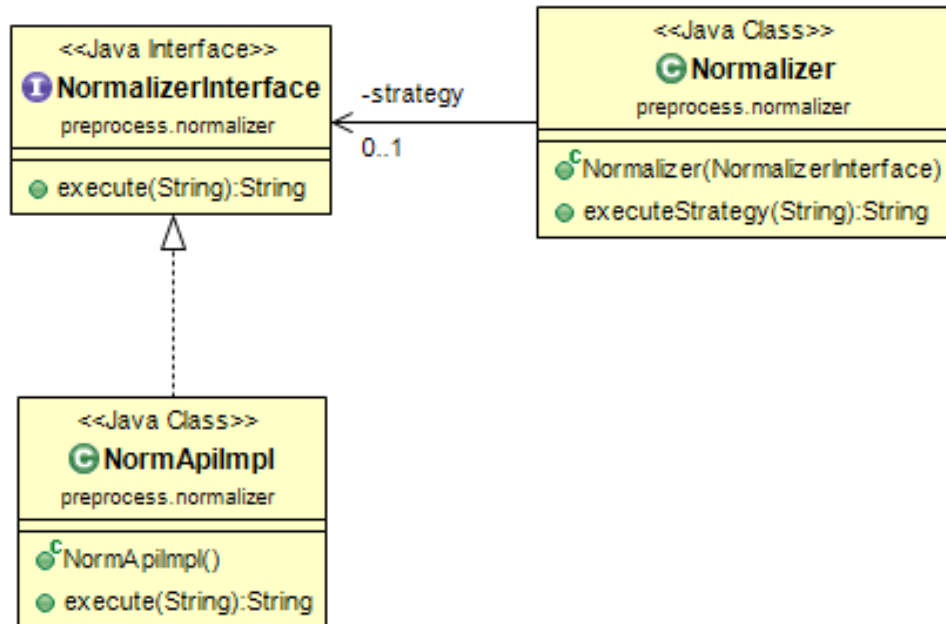


2.3.1 Preprocessor Manager

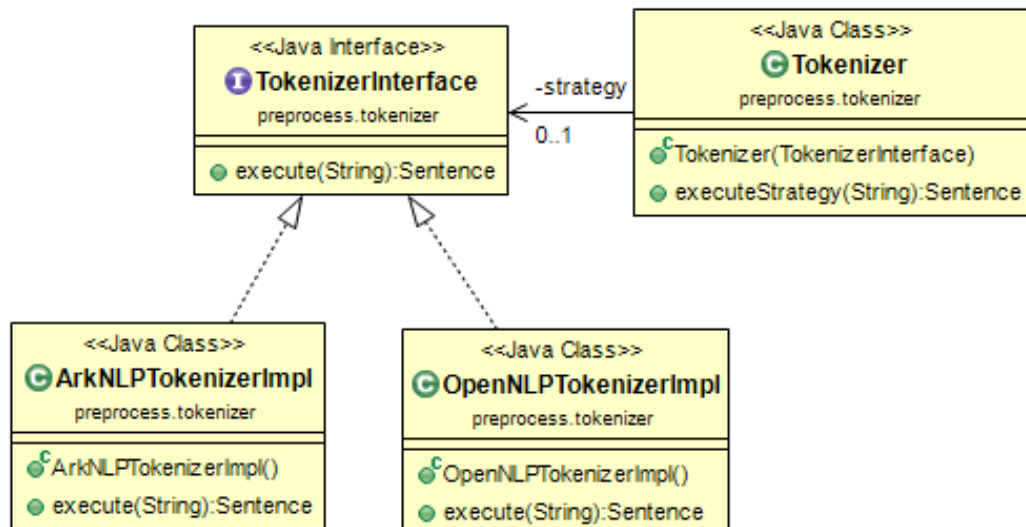




2.3.1.1 Normalizer

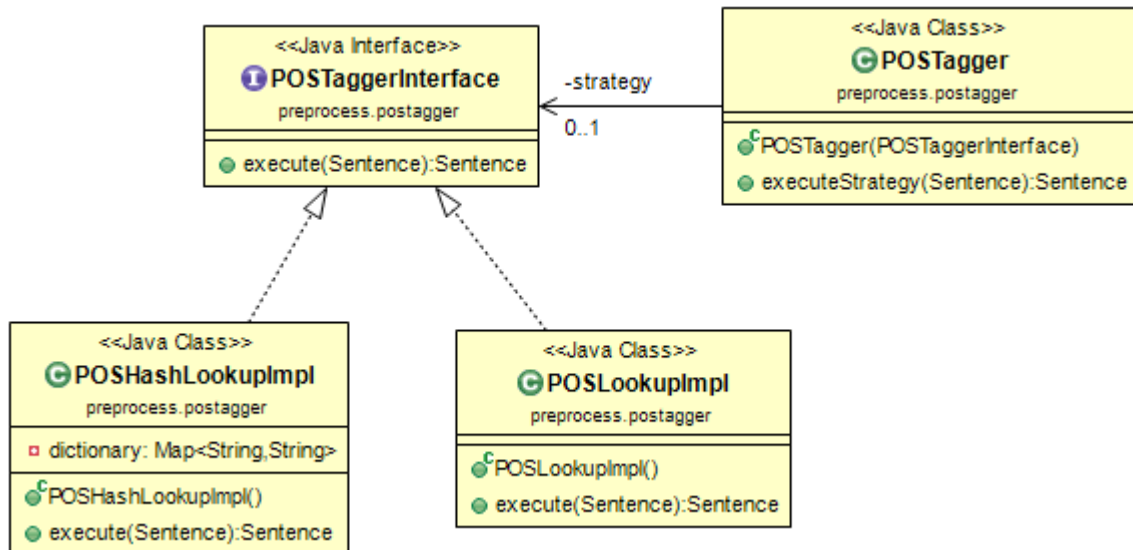


2.3.1.2 Tokenizer

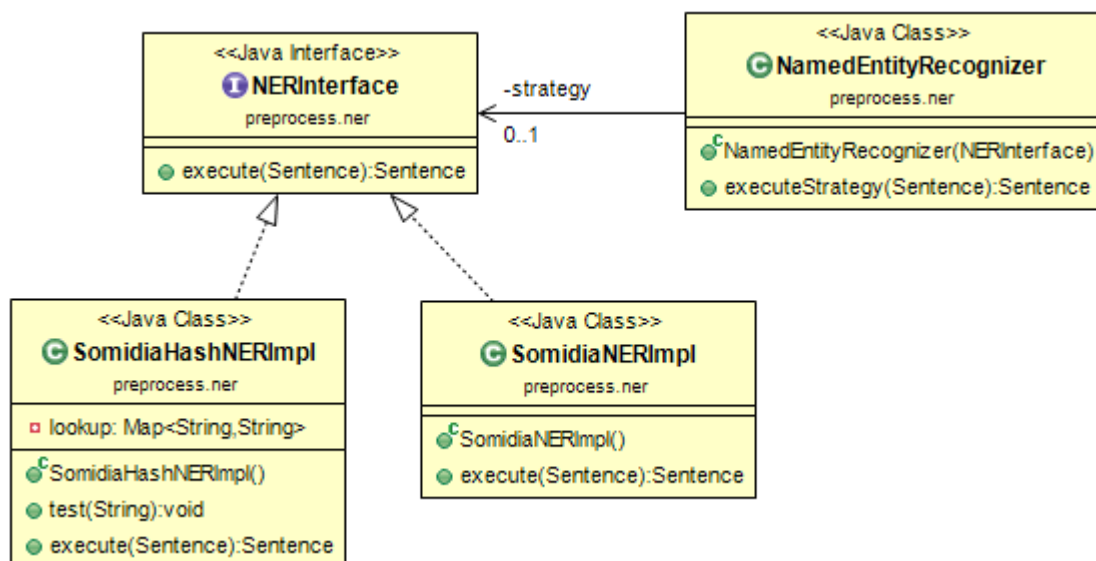




2.3.1.3 POS Tagger

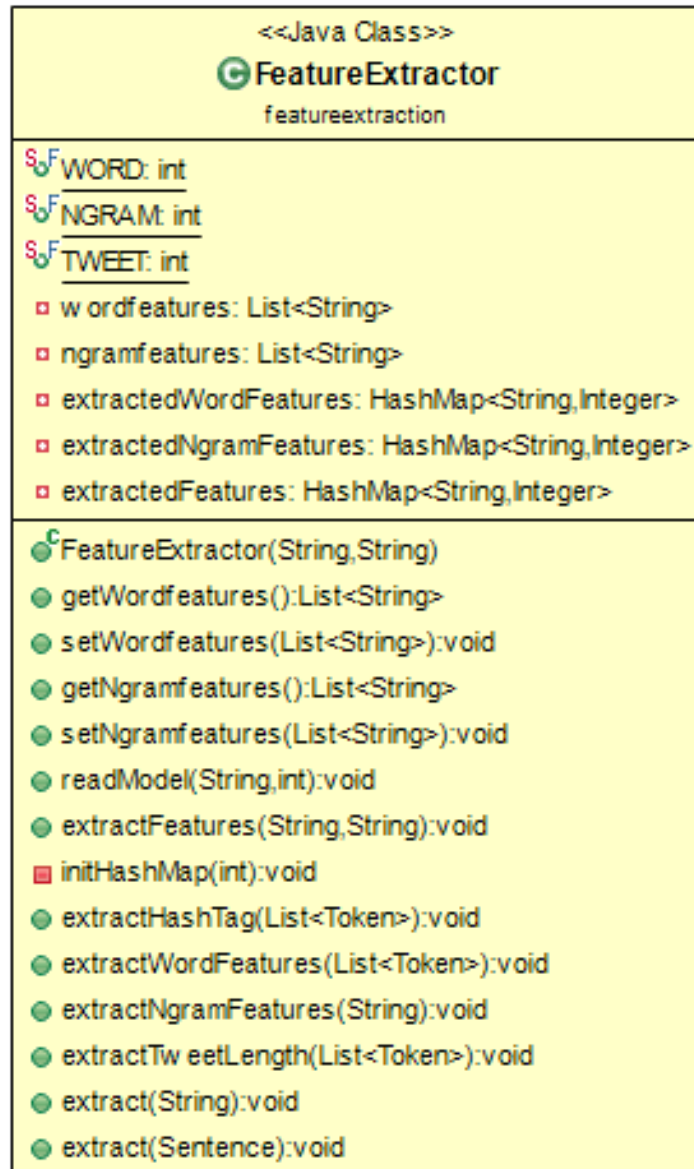


2.3.1.4 NER



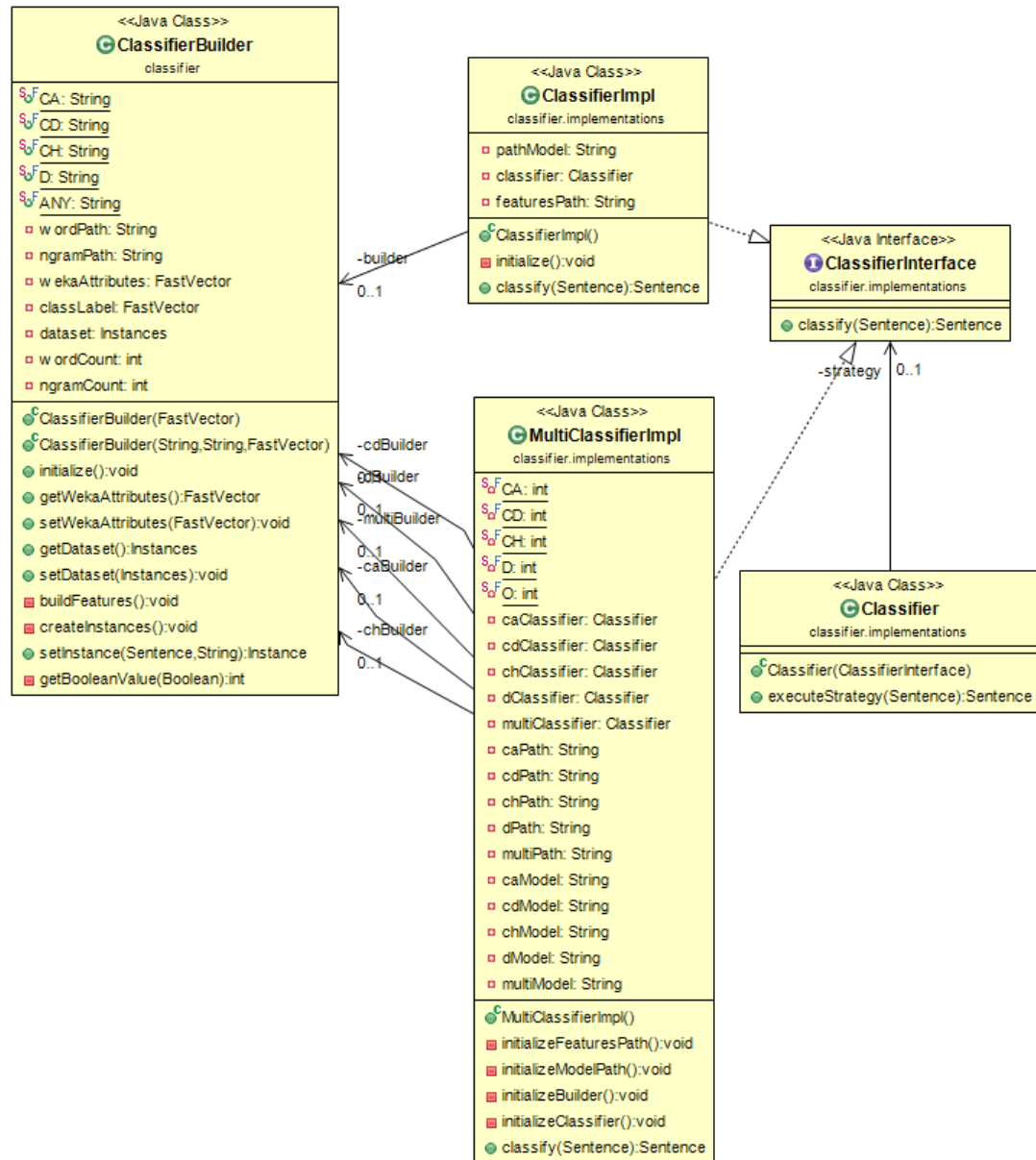


2.3.2 Feature Extraction Module



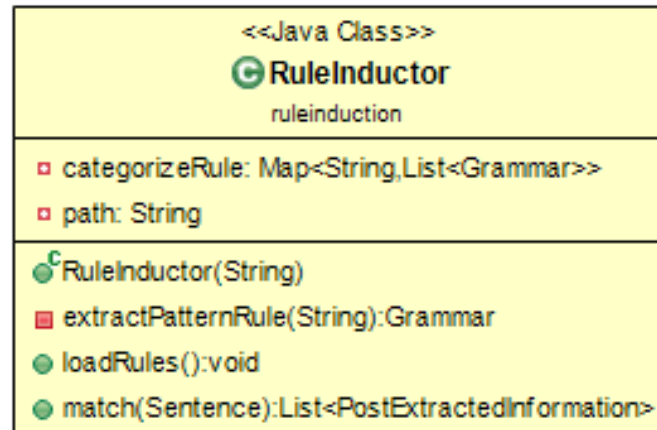


2.3.3 Classifier Module





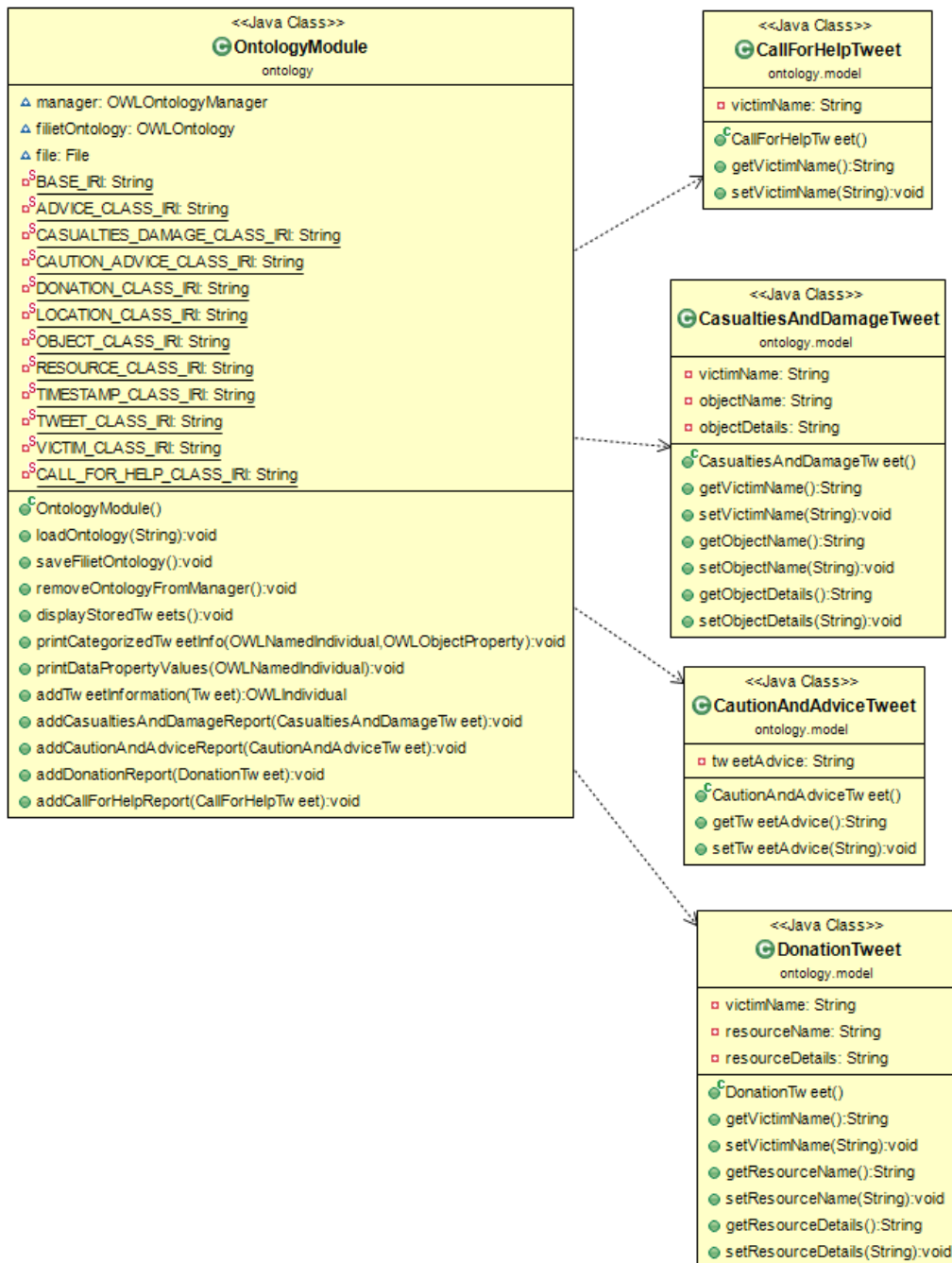
2.3.4 Rule Inductor Module





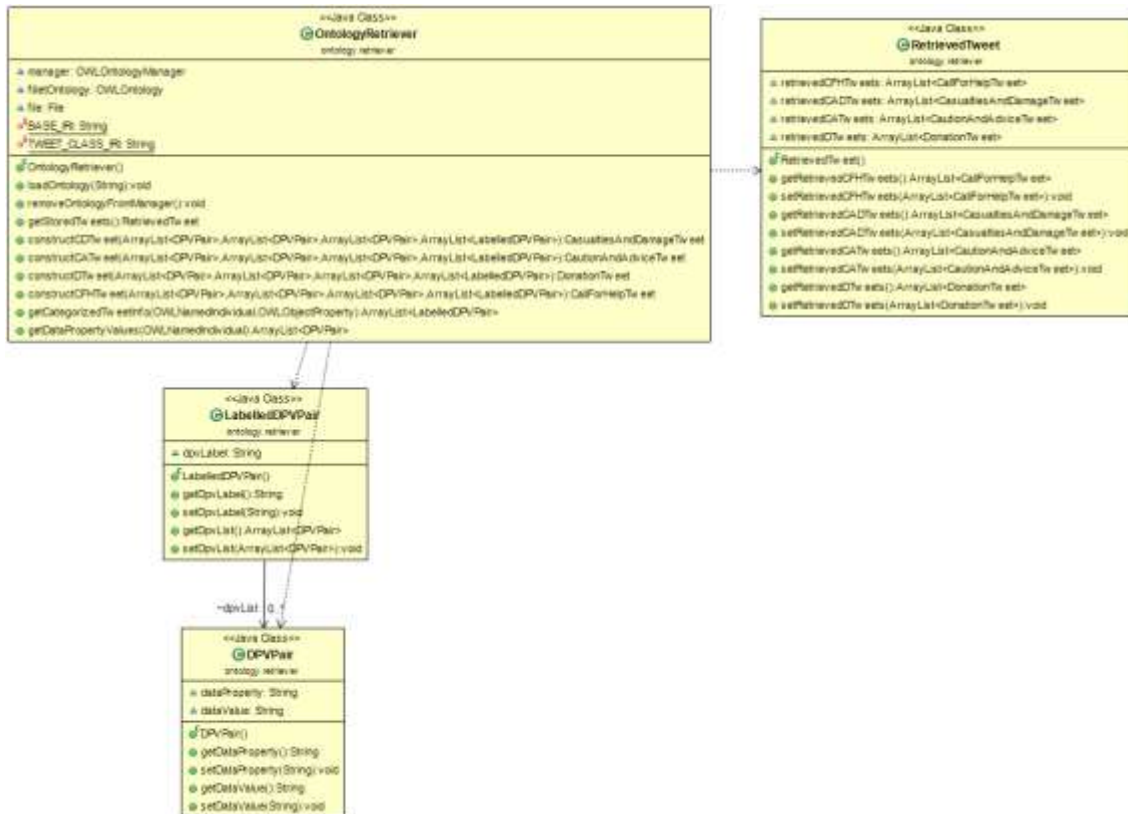
2.4 Ontology

2.4.1 OntologyModule

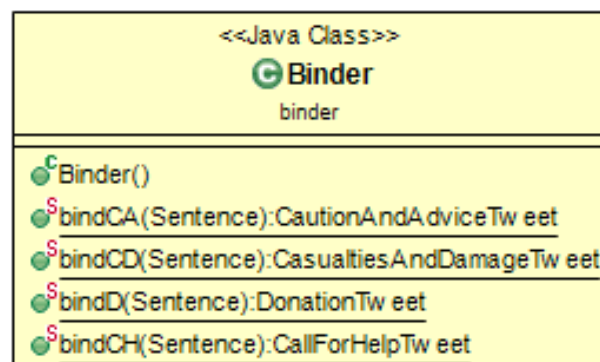




2.4.2 OntologyRetriever



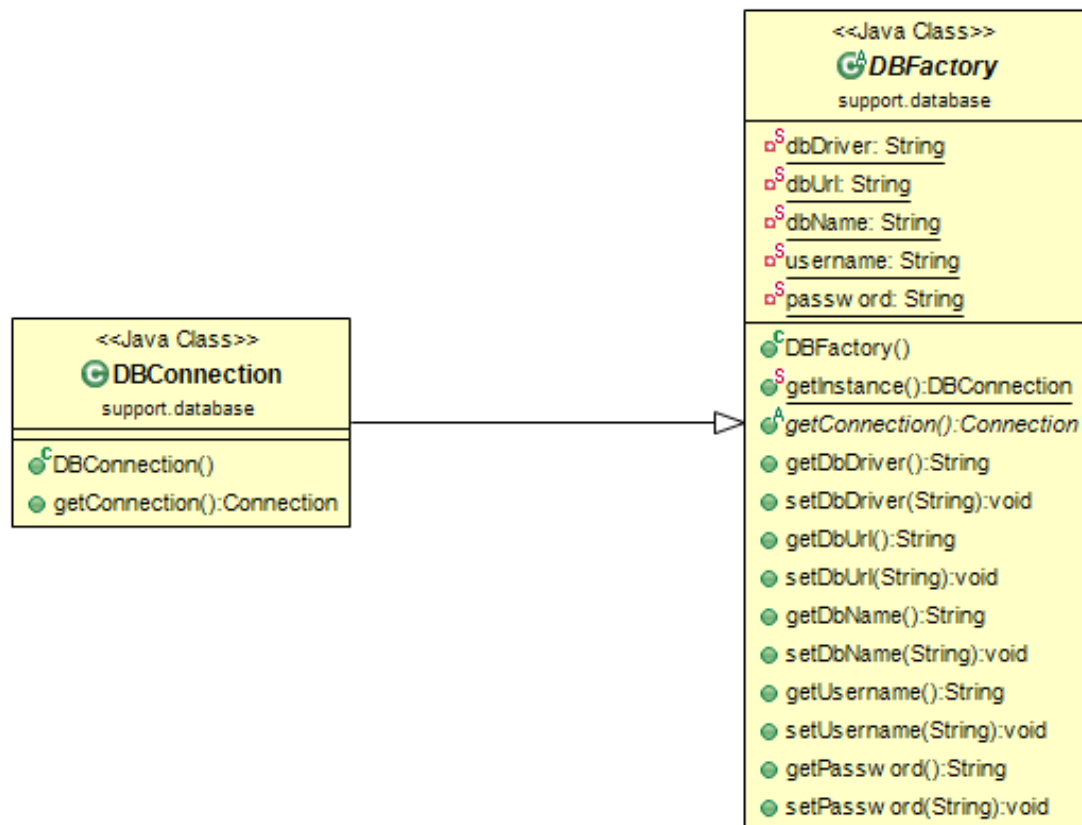
2.4.3 Binder



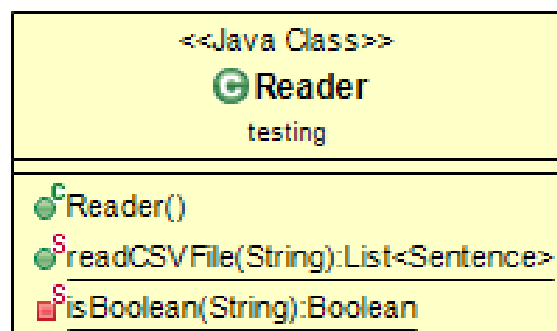


2.5 Others

2.5.1 Database



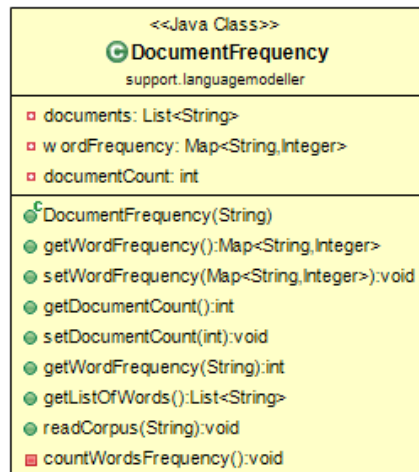
2.5.2 Reader



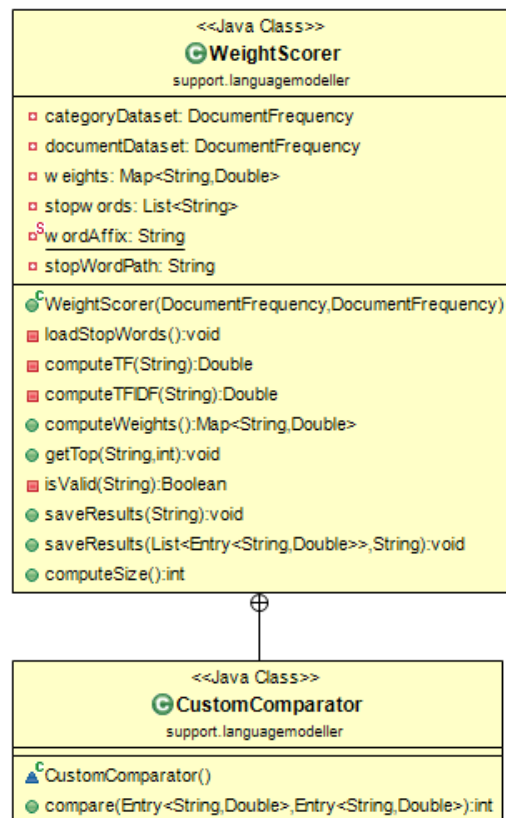


2.5.3 Language Modeller

2.5.3.1 Document Frequency

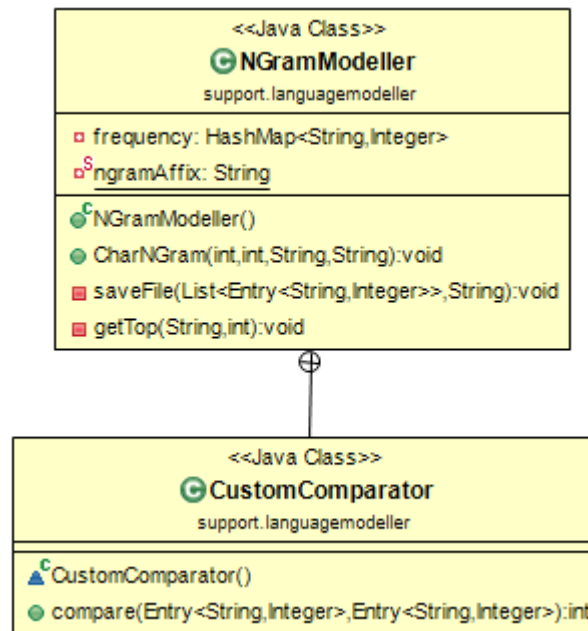


2.5.3.2 WeightScorer

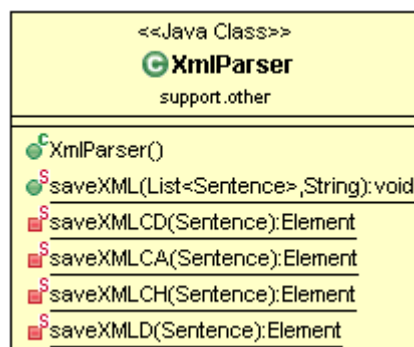




2.5.3.3 N-Gram Modeller



2.5.4 XML Parser





2.6 Entity Relationship Diagram





3.0 Class Dictionary

Module Name	File Name
Crawler Module	\filiet-repo\FILIET\resources\db.properties \filiet-repo\FILIET\resources\twitter.properties \filiet-repo\FILIET\resources\keywords.txt
Preprocessor Module	\filiet-repo\FILIET\resources\NamedEntityRecognizerDictModel \filiet-repo\FILIET\resources\posDictionary
Feature Extraction Module	\filiet-repo\FILIET\resources\model\TFIDF-Scores\mario-ruby\marioruby-word-combined-30.txt
Classifier Module	\filiet-repo\FILIET\resources\model\classifier\multiclassifier\mario-marioruby-randomforest-30-model.model \filiet-repo\FILIET\resources\model\classifier\multiclassifier\mario-rf-ca-30.model \filiet-repo\FILIET\resources\model\classifier\multiclassifier\mario-rf-cd-30.model \filiet-repo\FILIET\resources\model\classifier\multiclassifier\mario-rf-ch-30.model \filiet-repo\FILIET\resources\model\classifier\multiclassifier\mario-rf-d-30.model \filiet-repo\FILIET\resources\model\TFIDF-Scores\mario-ruby\marioruby-word-combined-30.txt \filiet-repo\FILIET\resources\model\TFIDF-Scores\mario\archive\mario-word-ca-30.txt \filiet-repo\FILIET\resources\model\TFIDF-Scores\mario\archive\mario-word-cd-30.txt \filiet-repo\FILIET\resources\model\TFIDF-Scores\mario\archive\mario-word-ch-30.txt \filiet-repo\FILIET\resources\model\TFIDF-Scores\mario\archive\mario-word-d-30.txt
Rule Inductor	\filiet-repo\FILIET\resources\rules\simple-rules
Ontology	\filiet-repo\FILIET\resources\ontology\Ruby_OWL.owl



3.1 Models

3.1.1 Tweet Class

Description	The Sentence class is the class that holds all information about the tweet instance
Type	Class

Attributes	Data Type	Constraint	Description
TweetID	Long	Private	Stores the ID of the tweet
User	String	Private	Stores the author of the tweet
Tweet	Double	Private	Stores the message (tweet)
Latitude	Double	Private	Stores the latitude where the author posted the tweet (if it is available)
Longitude	Double	Private	Stores the longitude where the author posted the tweet (if it is available)
Retweet	Boolean	Private	Tells if the tweet is a retweet
Hashtag	Boolean	Private	Tells if the tweet contains a hashtag
URL	Boolean	Private	Tells if the tweet contains URL link
Language	String	Private	Stores the language the tweet is written
Category	String	Private	Stores the manually annotated category
Status	Status	Private	Stores other information about the tweet (from Twitter4j library)

Method Name	Description	Parameters	Returns	Constraint
isURL	Tell if the tweet contains URL	Status (Tweet4j)	Boolean	Public
isHashtag	Tell if the tweet contains hashtag	Status (Tweet4j)	Boolean	Public
getTweetID	Return the tweet ID	Void	Long	Public
setTweetID	Sets the tweet ID	Long	Void	Public
getUser	Returns the	Void	String – tweet	Public



Method Name	Description	Parameters	Returns	Constraint
	tweet's author		user	
setUser	Se the tweet's author	String – tweet user	Void	Public
getTweet	Return the message (tweet)	Void	String - tweet	Public
setTweet	Sets the message (tweet)	String - tweet	Void	Public
getLatitude	Get the location (latitude)	Void	Double - latitude	Public
setLatitude	Set the location (latitude)	Double - latitude	Void	Public
getLongitude	Get the location (longitude)	Void	Double	Public
setLongitude	Set the location (longitude)	Double	Void	Public
getRetweet	Return the Retweet's value	Void	Boolean	Public
setRetweet	Set the Retweet value	Boolean	Void	Public
getHashtag	Get the Hashtag value	Void	Boolean	Public
setHashtag	Set the Hashtag value	Boolean	Void	Public
getURL	Get the URL value	Void	Boolean	Public
setURL	Set the URL value	Void	Boolean	Public
getLanguage	Get the language value	Void	String	Public



Method Name	Description	Parameters	Returns	Constraint
setLanguage	Set the language value	String	Void	Public
getCategory	Get the Category value	Void	String	Public
setCategory	Set the Category value	String	Void	Public
getStatus	Get the Status object (Twitter 4j)	Void	Status	Public
setStatus	Set the Status object (Twitter 4j)	Status	Void	Public
StoreTweet	Store the tweet in the database	Void	Void	Public

3.1.2 Sentence Class

Description	The Sentence class is the class that holds all information about the tweet
Type	Class

Attributes	Data Type	Constraint	Description
extractedWordFeatures	HashMap<String,Integer>	Private	Contains the extracted word features and its frequency.
extractedNgramFeatures	HashMap<String,Integer>	Private	Contains the extracted n-gram features and its frequency.
extractedFeatures	HashMap<String,Integer>	Private	Contains other features and its values (Length)
appliedRules	List<Grammar>	Private	Contains the list of extraction rules that was applied to the tweet.
tweets	Tweet	Private	Contains information about the tweet.
sentence	ArrayList<Token>	Private	Contains the



Attributes	Data Type	Constraint	Description
			tokenized tweet.
category	String	Private	The category labelled by the classifier.
extractedInformation	List<PostExtractedInformation>	Private	Contains the list of extracted information.

Method Name	Description	Parameters	Returns	Constraint
getTweets	Gets the Tweet information.	Void	Tweet – the tweet information	Public
setTweets	Set the Tweet information.	Tweet – the tweet information	Void	Public
getExtractedWordFeatures	Get the extracted word features	Void	HashMap<String, Integer> - contains the extracted words and its frequency	Public
setExtractedWordFeatures	Set the extracted word features	HashMap<String, Integer> - contains the extracted words and its frequency	Void	Public
getExtractedNgramFeatures	Get the extracted n-gram features	Void	HashMap<String, Integer> - contains the extracted n-grams and its frequency	Public
setExtractedNgramFeatures	Set the extracted n-gram features	HashMap<String, Integer> extracted n-grams and its frequency	Void	Public
getSentence	Get the ArrayList of Tokens (tokenized Tweet)	Void	ArrayList<Token> - tokenized tweets	Public
setSentence	Set the ArrayList of Tokens (tokenized Tweet)	ArrayList<Token> - tokenized tweet	Void	Public
addToken	Add a token to the ArrayList	Token – a single word	Void	Public



Method Name	Description	Parameters	Returns	Constraint
getLength	Get the length of the token	Void	Int – number of tokens	Public
getToken	Get the specific token with the same word	String – the word to find	Token – returns the token of the word	Public
getToken	Get the specific token using index	Int – index of the word	Token – returns the token with the indicated index	Public
getExtractedInformation	Get the extracted information	Void	List<PostExtractedInformation> - return the list of extracted information	Public
setExtractedInformation	Set the extracted information	List<PostExtractedInformation> - the list of extracted information	Void	Public
getCategory	Get the category labelled by the classifier	Void	String	Public
setCategory	Sets the category labelled by the classifier	String	Void	Public
getRawTweet	Gets the unprocessed tweet	Void	String – the unprocessed tweet	Public

3.1.3 Token Class

Description	The Token class holds information about a single word.
Type	Class

Attributes	Data Type	Constraint	Description
Word	String	Private	A tokenized word
POSTag	String	Private	The Part-of-Speech tag of the word
NERTag	String	Private	The entity of the word

Method Name	Description	Parameters	Returns	Constraint
getWord	Returns the	Void	String – the word	Public



Method Name	Description	Parameters	Returns	Constraint
	tokenized word			
setWord	Sets the tokenized word	String – the word	Void	Public
getPostTag	Returns the POS Tag of the word	Void	String – POS tag	Public
setPOSTag	Sets the POS Tag of the word	String – POS Tag	Void	Public
getNERTag	Returns the NER Tag of the word	Void	String – NER Tag	Public
setNERTag	Sets the NER Tag of the Word	String – NER tag	Void	Public
PrintToken	Prints the word and its POS Tags and NER Tags	Void	Void	Public

3.1.4 Rule Class

Description	The Rule class represents a token of the extraction rules.
Type	Class

Attributes	Data Type	Constraint	Description
type	String	Private	Tells what kind of extraction to match (POS, NER, or String)
value	String	Private	Tells the value that must be matched
asExtraction	String	Private	Tells the type of information that will be extracted
Sample Rule		<ner (type): LOCATION (value)>[as]LOCATION (asExtraction)	

Method Name	Description	Parameters	Returns	Constraint
getType	Return the type value	Void	String – the type of tag to be extracted	Public
setType	Set the type attribute	String – the type of tag to be extracted	Void	Public
getValue	Return the value value	Void	String – the value to be matched	Public
setValue	Set the value attribute	String – the value to be	Void	Public



Method Name	Description	Parameters	Returns	Constraint
		matched		
getAsExtraction	Return the asExtraction value	Void	String – the type of information that will be extracted	Public
setAsExtraction	Set the asExtraction attribute	String – the value to be matched	Void	Public
match	Checks if the token matches the rule.	Token – the token to be match	Boolean – returns true if it is a match, false if not.	Public

3.1.5 Grammar Class

Description	The Grammar class represents an extraction rules.
Type	Class

Attributes	Data Type	Constraint	Description
rules	List<Rules>	Private	Contains the list of Rule tokens

Method Name	Description	Parameters	Returns	Constraint
addRule	Add a Rule to the list	Rule - Rule token	Void	Public
getRules	Get the extraction rule	Void	List<Rule> - the extraction rule	Public
setRules	Set the extraction rule	List<Rule> - the extraction rule	Void	Public
getRuleCount	Get the number of Rule tokens	Void	Int – number of rule tokens	Public
printRules	Prints the extraction rules	Void	Void	Public

3.1.6 ExtractedInformation Class

Description	The ExtractedInformation Class represents a token of the extracted information.
Type	Class

Attributes	Data Type	Constraint	Description
informationType	String	Private	Tells the type of information the token is.



Attributes	Data Type	Constraint	Description
value	Token	Private	The token that was extracted

Method Name	Description	Parameters	Returns	Constraint
getInformationType	Return the informationType value	Void	String – the type of information	Public
setInformationType	Set the informationType attribute	String – the type of information	Void	Public
getValue	Return the value value	Void	Token	Public
setValue	Set the value attribute	Token	Void	Public

3.1.7 PostInformationExtracted Class

Description	The PostInformationExtracted Class represents the extracted information
Type	Class

Attributes	Data Type	Constraint	Description
type	String	Private	Stores the type of information
compiledInformation	List<ExtractedInformation>	Private	Stores the extracted information
appliedRules	Grammar	Private	Stores the rules that was used to extract the information

Method Name	Description	Parameters	Returns	Constraint
getAppliedRules	Returns the applied rules	Void	Grammar – the applied rule	Private
setAppliedRules	Set the applied rules	Grammar – the applied rule	Void	Private
getType	Get the type	Void	String	Private
setType	Set the type	String	Void	Private
getCompiledInformation	Get the compiledInformation	Void	List<ExtractedInformation>	Private
setCompiledInformation	Set the compiledInformation	List<ExtractedInformation>	Void	Private
printText	Prints the extracted information	Void	Void	Private



3.2 Crawler Module

3.2.1 Crawler Class

Description	The Crawler Class collects the Tweets from Twitter.
Type	Class

Attributes	Data Type	Constraint	Description
ConsumerKey	String	Private	This is used to allow the API to make calls to Twitter.
ConsumerSecret	String	Private	This is used to allow the API to make calls to Twitter.
AccessKey	String	Private	This is the OAuth token to represent the user.
AccessSecret	String	Private	This is the OAuth token to represent the user.
twitterStreamFactory	TwitterStreamFactory (Twitter4j)	Private	Generates the Twitter streams

Method Name	Description	Parameters	Returns	Constraint
InitializeKey	Initializes the ConsumerKey, Consumer Secret, AccessKey, and AccessSecret	Void	Void	Private
InitializeTwitter	Initializes the TwitterStreamFactory	Void	Void	Private
CrawlTweet	Listens to incoming tweets and store them in the database	Void	Void	Public

3.3 Information Extraction Engine

3.3.1 InformationEngineExtraction Class

Description	The InformationEngineExtraction class integrates the preprocessing module, feature extraction module, classifier module, and the rule inductor module.
Type	Class



Attributes	Data Type	Constraint	Description
word	String	Private	Path to the word features file
rulePath	String	Private	Path to the rule list file
preprocessor	PreprocessorManager	Private	Handles all the preprocessing of the tweets
feature	FeatureExtractor	Private	Extracts the features in the tweet
classifier	Classifier	Private	Classifies the tweet into the category
ruleInductor	RuleInductor	Private	Applies extraction rules to get the relevant information

Method Name	Description	Parameters	Returns	Constraint
runExtractor	Runs all the extraction modules. Returns a processed tweet.	String – the tweet	Sentence – processed tweet	Public
setPreprocessor	Sets the PreprocessorManager	PreprocessorManager – the preprocessor module	Void	Public
setFeature	Sets the FeatureExtractor	FeatureExtractor – the feature extractor module	Void	Public
setClassifier	Sets the Classifier	Classifier – the classifier module	Void	Public
setRuleInductor	Sets the RuleInductor	RuleInductor – the rule inductor	Void	Public

3.4 Preprocessing Module

3.4.1 Preprocessor Manager Class

Description	The Preprocessor Manager handles all the preprocessing module
Type	Class

Attributes	Data Type	Constraint	Description
normalizer	Normalizer	Private	Normalizer module
tokenizer	Tokenizer	Private	Tokenizer module
post	POSTagger	Private	POS Tagger module



Attributes	Data Type	Constraint	Description
ner	NamedEntityRecognizer	Private	NER module

Method Name	Description	Parameters	Returns	Constraint
InitializeModule	Initializes all the preprocessing modules	Void	Void	Private
PreprocessText	The raw tweet will be processed through the preprocessing modules	String – the tweet	Sentence – preprocessed tweet	Public

3.4.2 Normalizer

3.4.2.1 NormalizerInterface Class

Description	The NormalizerInterface class is an interface that tells to implement the executeStrategy() method.
Type	Interface

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the normalizer	String – raw tweet	Sentence – normalized tweet	Public

3.4.2.2 Normalizer Class

Description	The Normalizer class is the class that implements the NormalizerInterface.
Type	Class

Attributes	Data Type	Constraint	Description
strategy	NormalizerInterface	Private	Stores the implementation that will be used by the executeStrategy method

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the	String – raw	String –	Public



Method Name	Description	Parameters	Returns	Constraint
	normalizer. The Normalizer returns the tweet that has been normalized	tweet	normalized tweet	

3.4.2.3 NormApiImpl Class

Description	The NormApiImpl class implements the Tokenizer using the NormAPI
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the normalizes that uses Norm API	String – raw tweet	String – normalized tweet	Public

3.4.3 Tokenizer

3.4.3.1 TokenizerInterface Class

Description	The TokenizerInterface class is an interface that tells to implement the executeStrategy() method.
Type	Interface

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the tokenizer	String – normalized tweet	Sentence – tokenized tweet	Public

3.4.3.2 Tokenizer Class

Description	The Tokenizer class is the class that implements the TokenizerInterface.
Type	Class



Attributes	Data Type	Constraint	Description
strategy	TokenizerInterface	Private	Stores the implementation that will be used by the executeStrategy method

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the tokenizer. Returns a Sentence object that has the tokenized tweet.	String – normalized tweet	Sentence – tokenized tweet	Public

3.4.3.3 ArkNLPTokenizer Class

Description	The ArkNLPTokenizer class implements the Tokenizer using the ArkNLP library.
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the tokenizer that uses the ArkNLP Tokenizer	String – normalized tweet	Sentence – tokenized tweet	Public

3.4.3.4 OpenNLPTokenizer Class

Description	The OpenNLPTokenizer class implements the Tokenizer using the OpenNLP library.
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the tokenizer that uses OpenNLP Tokenizer	String – normalized tweet	Sentence – tokenized tweet	Public



3.4.4 POS Tagger

3.4.4.1 POSInterface Class

Description	The POSInterface class is an interface that tells to implement the executeStrategy() method.
Type	Interface

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the POS Tagger	Sentence – tokenized tweet	Sentence – POS tagged	Public

3.4.4.2 POSTagger Class

Description	The POSTagger class is the class that implements the POSInterface.
Type	Class

Attributes	Data Type	Constraint	Description
strategy	POSInterface	Public	Stores the implementation that the executeStrategy will use

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the POS Tagger. Each token now may contain a POS tag.	Sentence – tokenized tweet	Sentence – POS tagged	Public

3.4.4.3 POSHashLookupImpl Class

Description	The POSHashLookupImpl class implements POS lookup that uses a HashMap
Type	Class

Attributes	Data Type	Constraint	Description
dictionary	HashMap<String,Integer>	Private	Stores the list of words and its corresponding part-of-speech



Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the POS Tagger using the POS Hash Lookup implementation	Sentence – tokenized tweet	Sentence – POS tagged	Public

3.4.4.4 POSLookupImpl Class

Description	The POSLookupImpl class implements POS lookup.
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the POS Tagger using POS Lookup Implementation	Sentence – tokenized tweet	Sentence – POS tagged	Public

3.4.5 NER

3.4.5.1 NERInterface Class

Description	The NERInterface class is an interface that tells to implement the executeStrategy() method.
Type	Interface

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the NER Tagger. Each token now may contain a NER tag.	Sentence – POS Tagged	Sentence – NER tagged	Public



3.4.5.2 NamedEntityRecognizer Class

Description	The NamedEntityRecognizer class is the class that implements the NERInterface.
Type	Class

Attributes	Data Type	Constraint	Description
strategy	NERInterface	Public	Stores the implementation that the executeStrategy will use

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the POS Tagger. Each token now may contain a POS tag.	Sentence – POS tagged	Sentence – NER tagged	Public

3.4.5.3 SomidiaHashNERImpl Class

Description	The SomidiaHashNERImpl class implements NER lookup that uses a HashMap
Type	Class

Attributes	Data Type	Constraint	Description
lookup	HashMap<String,Integer>	Private	Stores the list of words and its corresponding entities

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the NER Tagger that uses SOMIDIA Hash Lookup Implementation.	Sentence – POS tagged	Sentence – NER tagged	Public

3.4.5.4 SomidiaNERImpl Class

Description	The SomidiaHashNERImpl class implements NER lookup.
Type	Class

Attributes	Data Type	Constraint	Description



Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the NER Tagger that uses SOMIDIA Lookup Implementation	Sentence – POS tagged	Sentence – NER tagged	Public

3.5 Feature Extractor Module

3.5.1 FeatureExtraction Class

Description	The FeatureExtraction Class handles the feature extraction module.
Type	Class

Attributes	Data Type	Constraint	Description
wordfeatures	List<String>	Private	For CSV headers (word features)
ngramfeatures	List<String>	Private	For CSV headers (n-gram features)
extractedWordfeatures	HashMap<String,Integer>	Private	Stores the extracted word features
extractedNgramFeatures	HashMap<String,Integer>	Private	Stores the extracted n-gram features
extractedFeatures	HashMap<String,Integer>	Private	Stores the extracted other (tweet length) features

Method Name	Description	Parameters	Returns	Constraint
getWordFeatures	Return the wordfeatures	Void	List<String> - word features	Public
setWordFeatures	Set the wordfeatures	List<String> - word features	Void	Public
getNgramFeatures	Return the ngramfeatures	Void	List<String> - ngram features	Public
setNgramFeatures	Set the ngramfeatures	List<String> - ngram features	Void	Public
readModel	Reads the text file that contained the features to be used	String – path to the features, type – tells if it's a word features list or n-gram feature list	Void	Public
extractWordFeatures	Extract the word features from a tokenized text	List<Token> - tokenized	Void	Public
extractNgramFeatures	Extracts the n-	String -	Void	Public



Method Name	Description	Parameters	Returns	Constraint
atures	gram features from a text	untokenized		
extractTweetLength	Count the length of the tweet	List<Token> - tokenized	Void	Public
extract	Extracts the unprocessed instance (used for batch processing)	String – normalized tweet	Void	Public
extract	Extracts the feature of a processed instance	Sentence – preprocessed tweet	Sentence – features extracted	Public
extractFeatures	Extracts the features of a corpus and save the result into a CSV	String – path, String – save	Void	Public

3.6 Classifier Module

3.6.1 ClassifierInterface Class

Description	The ClassifierInterface class is an interface that tells to implement the classify() method.
Type	Interface

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
classify	Executes the classifier	Sentence – feature extracted	Sentence – classified tweet	Public

3.6.2 ClassifierBuilder Class

Description	The ClassifierBuilder Class creates an Instance class that will be used to classify. It sets the values from the feature extraction to the Instance class.
Type	Class

Attributes	Data Type	Constraint	Description



Method Name	Description	Parameters	Returns	Constraint
Initialize	Initialize the models that will be used for the classification	Void	Void	Public
getWekaAttributes	Get the list of attributes	Void	FastVector<String> (Weka) - the features that the classifier will use	Public
setWekaAttributes	Set the list of attributes	FastVector<String> (Weka) - the features that the classifier will use	Void	Public
getDataset	Get the created Instances class	Void	Instances (Weka) – the header for the Instance	Public
setDataset	Set an Instances class	Instances (Weka) – the header for the Instance	Void	Public
buildFeatures	Sets the attributes that will be used for classification	Void	Void	Private
createInstance	Creates the Instances class	Void	Void	Private
setInstance	Set the values for the Instance object	Sentence – the tweet to be classified, String – the type of classifier	Void	Public
getBooleanValue	Returns 1 if true, else 0.	Boolean	Int	Private

3.6.3 Classifier Class

Description	The Classifier Class implements the ClassifierInterface.
Type	Class

Attributes	Data Type	Constraint	Description
strategy	ClassifierInterface	Private	Stores the implementation that will be used by the executeStrategy



			method
--	--	--	--------

Method Name	Description	Parameters	Returns	Constraint
classify	Executes the classify method. The Classifier sets the Category field	Sentence – the feature extracted tweet	Sentence – the classified tweet	Public

3.6.4 ClassifierImpl Class

Description	The ClassifierImpl Class is a classifier that uses a single classifier.
Type	Class

Attributes	Data Type	Constraint	Description
pathModel	String	Private	Path to the classifier model
classifier	Classifier (Weka)	Private	Classifier that classifies all categories
featuresPath	String	Private	Path to the features
Builder	ClassifierBuilder	Private	Creates the instance for the classifier

Method Name	Description	Parameters	Returns	Constraint
Initialize	Initializes the Classifier	Void	Void	Private
executeStrategy	Executes the classify method that uses a single classifier implementation.	Sentence – the feature extracted tweet	Sentence – the classified tweet	Public

3.6.5 MultiClassifierImpl Class

Description	The MultiClassifierImpl Class is a classifier that implements a multiple classifier. The classifier is consist of 5 classifiers, 4 binary classifier for each category, and 1 single classifier.
Type	Class

Attributes	Data Type	Constraint	Description
caClassifier	Classifier (Weka)	Private	Classifier that classifies CA and O
cdClassifier	Classifier (Weka)	Private	Classifier that



Attributes	Data Type	Constraint	Description
			classifies CD and O
chClassifier	Classifier (Weka)	Private	Classifier that classifies CH and O
dClassifier	Classifier (Weka)	Private	Classifier that classifies D and O
multiClassifier	Classifier (Weka)	Private	Classifier that classifies all categories
caPath	String	Private	Path to the CA features
cdPath	String	Private	Path to the CD features
chPath	String	Private	Path to the CH features
dPath	String	Private	Path to the D features
multiPath	String	Private	Path to the combined features
caModel	String	Private	Path to the CA model
cdModel	String	Private	Path to the CD model
chModel	String	Private	Path to the CH model
dModel	String	Private	Path to the D model
multiModel	String	Private	Path to the combined model
caBuilder	ClassifierBuilder	Private	Creates the instance for the caClassifier
cdBuilder	ClassifierBuilder	Private	Creates the instance for the cdClassifier
chBuilder	ClassifierBuilder	Private	Creates the instance for the chClassifier
dBuilder	ClassifierBuilder	Private	Creates the instance for the dClassifier
multiBuilder	ClassifierBuilder	Private	Creates the instance for the multiClassifier

Method Name	Description	Parameters	Returns	Constraint
executeStrategy	Executes the classify method that uses the multiple classifier implementation.	Sentence – the feature extracted tweet	Sentence – the classified tweet	Public
initializeFeaturesPath	Sets the location of the features that will be used	Void	Void	Private
initializeModelPath	Sets the location of the models that will be used	Void	Void	Private
initializeBuilder	Initializes the	Void	Void	Private



Method Name	Description	Parameters	Returns	Constraint
	ClassifierBuilder class			
initializeClassifier	Initializes the Classifiers.	Void	Void	Private

3.7 Rule Induction Module

Description	The RuleInductor Class extracts the information from the tweets by matching the rules.
Type	Class

Attributes	Data Type	Constraint	Description
categorizeRule	Map<String,List<Grammar>>	Private	Contains the list of rules for each category
path	String	Private	Path to the rules

Method Name	Description	Parameters	Returns	Constraint
loadRules	Loads the rules from the text files	Void	Void	Public
extractPatternRules	Parse the rules from the text files	String – path file to the rules	Void	Private
match	Extracts the information by matching the rules	Sentence - classified tweet	List<PostExecuteInformation> - extracted information	Public

3.8 Ontology Module

3.8.1 OntologyModule Class

Description	The OntologyModule class is responsible for adding new instances into the ontology.
Type	Class

Attributes	Data Type	Constraint	Description
manager	OwlOntologyManager	Default	The ontology uses a an OwlManager variable to facilitate the manipulation of the ontology file.
filietOntology	OwlOntology	Default	This is the variable that contains the actual OWL file of the



Attributes	Data Type	Constraint	Description
			ontology.
File	File	Default	This is just a variable for containing the OWL file that will be manipulated in the system.
BASE_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the actual ontology used by the system.
ADVICE_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Advice class in the actual ontology used by the system.
CASUALTIES_DAMAGE_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Casualties and Damage class in the actual ontology used by the system.
CAUTION_ADVICE_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Caution and Advice class in the actual ontology used by the system.
DONATION_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Donation class in the actual ontology used by the system.
LOCATION_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Location class in the actual ontology



Attributes	Data Type	Constraint	Description
			used by the system.
OBJECT_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Object class in the actual ontology used by the system.
TIMESTAMP_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Timestamp class in the actual ontology used by the system.
TWEET_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Tweet class in the actual ontology used by the system.
VICTIM_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Victim class in the actual ontology used by the system.
CALL_FOR_HELP_CLASS_IRI	String	Private	This is a string that contains the Internationalized Resource Identifier of the Call For Help class in the actual ontology used by the system.

Method Name	Description	Parameters	Returns	Constraint
loadOntology	Loads the ontology file to the system's ontology manager so that it can be manipulated by the system.	Void	Void	Public
saveFilietOntolo	Save the	Void	Void	Public



Method Name	Description	Parameters	Returns	Constraint
gy	changes that have been made to the ontology.			
removeOntologyFromManager	Removes the ontology from the manager to prevent unwanted changes.	Void	Void	Public
displayStoredTweets	Display the content of the ontology	Void	Void	Public
printCategorizedTweetInfo	This is for checking purposes so that specific information about the tweets that were stored to the ontology can be viewed.	OWLNamedIndividual, OWLObjectProperty	Void	Public
printDataPropertyValues	This prints out the value of the specified data properties.	OWLNamedIndividual	Void	Public
addTweetInformation	Adds information that are related to the tweets like Location and Timestamp.	Tweet – holds information about the tweet	Void	Public
addCasualtiesAndDamageReport	Adds a new instance of a Casualties and Damage tweets into the ontology.	CasualtiesAndDamageTweet – holds information about Casualties and Damage	Void	Public
addDonationReport	Adds a new instance of a Donation tweets into the ontology.	DonationTweet – holds information about Donations	Void	Public
addCallForHelpReport	Adds a new instance of a Call For Help tweets into the ontology.	CallForHelpTweet – hold information about Call for Help	Void	Public
addCautionAndAdviceReport	Adds a new instance of a Caution and Advice tweets	CautionAndAdviceTweet – holds information about Caution	Void	Public



Method Name	Description	Parameters	Returns	Constraint
	into the ontology.	and Advice		

3.8.2 Tweet Class (Ontology)

Description	The Tweet (Ontology) Class represents the Tweet information in the ontology
Type	Class

Attributes	Data Type	Constraint	Description
tweetHandle	String	Private	Stores the tweet's author
tweetContent	String	Private	Stores the message (tweet)
tweetGeoLocation	String	Private	Stores the location where the tweet was sent (latitude, longitude)
locationInTweet	String	Private	Stores the location that was extracted in the tweet
tweetTimestamp	String	Private	Stores the tweet's timestamp
tweetDate	String	Private	Stores the tweet's date

Method Name	Description	Parameters	Returns	Constraint
getTweetHandle	Get the tweet handle	Void	String	Public
setTweetHandle	Set the tweet handle	String	Void	Public
getTweetContent	Get the tweet content	Void	String - tweet	Public
setTweetContent	Set the tweet content	String - tweet	Void	Public
getTweetGeoLocation	Get the tweet's geolocation (latitude, longitude)	Void	String - latitude and longitude (Format: Latitude, Longitude)	Public
setTweetGeoLocation	Set the tweet's geolocation (latitude, longitude)	String – latitude and longitude (Format: Latitude, Longitude)	Void	Public
getLocationInTw	Get the	Void	String –	Public



Method Name	Description	Parameters	Returns	Constraint
getLocationInTweet	locationInTweet		extracted location in tweet	
setLocationInTweet	Set the locationInTweet	String – extracted location in tweet	Void	Public
getTweetTimestamp	Get the tweet timestamp	Void	String – timestamp of the tweet	Public
setTweetTimestamp	Set the tweet timestamp	String – timestamp of the tweet	Void	Public
getTweetDate	Get the tweet date	Void	String – date of the tweet	Public
setTweetDate	Set the tweet date	String – date of the tweet	Void	Public

3.8.3 CallForHelpTweet Class

Description	The CallForHelpTweet Class represents the Call For Help category in the ontology.
Type	Class
Inherited	Tweet (Ontology) Class

Attributes	Data Type	Constraint	Description
victimName	String	Private	Stores the extracted victim name in the tweet.

Method Name	Description	Parameters	Returns	Constraint
getVictimName	Get the victim name	Void	String – extracted victim name	Public
setVictimName	Set the victim name	String – extracted victim name	Void	Public

3.8.4 CasualtiesAndDamageTweet Class

Description	The CasualtiesAndDamageTweet Class represents the Casualties and Damage category in the ontology.
Type	Class
Inherited	Tweet (Ontology) Class



Attributes	Data Type	Constraint	Description
victimName	String	Public	Stores the extracted victim name in the tweet.
objectName	String	Public	Stores the extracted object that was destroyed in the tweet
objectDetails	String	Public	Stores the detail about the object

Method Name	Description	Parameters	Returns	Constraint
getVictimName	Get the victim name	Void	String – extracted victim name	Public
setVictimName	Set the victim name	String – extracted victim name	Void	Public
getObjectName	Get the object name	Void	String – extracted object	Public
setObjectName	Set the object name	String – extracted object	Void	Public
getObjectDetails	Get the object details	Void	String – extracted detail about the object	Public
setObjectDetails	Set the object details	String – extracted detail about the object	Void	Public

3.8.5 CautionAndAdviceTweet Class

Description	The CautionAndAdviceTweet Class represents the Caution and Advice category in the ontology.
Type	Class
Inherited	Tweet (Ontology) Class

Attributes	Data Type	Constraint	Description
tweetAdvice	String	Private	Stores the extracted advice in the tweet

Method Name	Description	Parameters	Returns	Constraint
getTweetAdvice	Get the tweet advice	Void	String – extracted advice	Public
setTweetAdvice	Set the tweet advice	String – extracted advice	Void	Public



3.8.6 DonationTweet Class

Description	The DonationTweet Class represents the Donation category in the ontology.
Type	Class
Inherited	Tweet (Ontology) Class

Attributes	Data Type	Constraint	Description
victimName	String	Public	Stores the extracted victim name in the tweet.
resourceName	String	Public	Stores the extracted resource that was donated in the tweet
resourceDetail	String	Public	Stores the details about the resource

Method Name	Description	Parameters	Returns	Constraint
getVictimName	Get the victim name	Void	String – extracted victim	Public
setVictimName	Set the victim name	String – extracted victim	Void	Public
getResourceName	Get the resource name	Void	String – extracted resource	Public
setResourceName	Set the resource name	String – extracted resource	Void	Public
getResourceDetail	Get the resource detail	Void	String – extracted resource detail	Public
setResourceDetail	Set the resource detail	String – extracted resource detail	Void	Public

3.8.7 OntologyRetriever Class

Description	The RuleInductor Class extracts the information from the tweets by matching the rules.
Type	Class

Attributes	Data Type	Constraint	Description



Method Name	Description	Parameters	Returns	Constraint
loadOntology	Loads the ontology file to the system's ontology manager so that it can be manipulated by the system.	Void	Void	Public
removeOntologyFromManager	Removes the ontology file from the manager so that no unwanted changes can be made to the ontology.	Void	Void	Public
getStoredTweets	Retrieves the tweets stored in the ontology.	Void	RetrievedTweet – the tweet in the ontology	Public
constructCDTweet	Constructs the CD tweet based from the different information extracted so that it can be manipulated within the system.	ArrayList<DPVP air>, ArrayList<DPVP air>, ArrayList<DPVP air>, ArrayList<LabelledDPVPair>	CasualtiesAndDamageTweet	Public
constructCATweet	Constructs the CA tweet based from the different information extracted so that it can be manipulated within the system.	ArrayList<DPVP air>, ArrayList<DPVP air>, ArrayList<DPVP air>, ArrayList<LabelledDPVPair>	CautionAndAdviceTweet	Public
constructCHTweet	Constructs the CH tweet based from the different information extracted so that it can be manipulated within the system.	ArrayList<DPVP air>, ArrayList<DPVP air>, ArrayList<DPVP air>, ArrayList<LabelledDPVPair>	CallForHelpTweet	Public
constructDTweet	Constructs the D tweet based	ArrayList<DPVP air>,	DonationTweet	Public



	from the different information extracted so that it can be manipulated within the system.	ArrayList<DPVP air>, ArrayList<DPVP air>, ArrayList<LabelledDPVPair>		
--	---	--	--	--

3.8.8 DPVPair Class

Description	This is a class that stores a pair of data properties and their respective values as they are retrieved from the ontology. This is a simple implementation to minimize the need for using hashmaps for accessing this types of data.
Type	Class

Attributes	Data Type	Constraint	Description
dataProperty	String	Default	This string contains the name of the data property.
dataValue	String	Default	This string contains the actual value for the specified data property.

Method Name	Description	Parameters	Returns	Constraint
getDataProperty	Get the data property	Void	String – data property	Public
setDataProperty	Set the data property	String – data property	Void	Public
getDataValue	Get the data value	Void	String – value of the data property	Public
setDataValue	Set the data value	String – value of the data property	Void	Public

3.8.9 LabelledDPVPair Class

Description	This class is for labelling the list of data property value pairs.
Type	Class

Attributes	Data Type	Constraint	Description
dpvLabel	String	Default	This contains the name of the list of data property value pairs.



Attributes	Data Type	Constraint	Description
dpvList	ArrayList<DPVPair>	Default	This is a list of data property value pairs.

Method Name	Description	Parameters	Returns	Constraint
getDpvLabel	Get the DPV Label	Void	String	Public
setDpvLabel	Set the DPV Label	String	Void	Public
getDpvList	Get the DPV List	Void	ArrayList<DPVPair>	Public
setDpvList	Set the DPV List	ArrayList<DPVPair>	Void	Public

3.8.10 RetrievedTweet Class

Description	This is a class that compiles the different arraylist of the tweets that were retrieved from the ontology.
Type	Class

Attributes	Data Type	Constraint	Description
retrievedCFHTweets	ArrayList<CallForHelpTweet>	Default	This is the ArrayList that contains tweet contains Call For Help tweet instances.
retrievedCADTweets	ArrayList<CasualtiesAndDamageTweet>	Default	This is the ArrayList that contains tweet contains Casualties and Damage tweet instances.
retrievedCATweets	ArrayList<CautionAndAdviceTweet>	Default	This is the ArrayList that contains tweet contains Caution and Advice tweet instances.
retrievedDtweets	ArrayList<DonationTweet>	Default	This is the ArrayList that contains tweet contains Donation tweet instances.

Method Name	Description	Parameters	Returns	Constraint
getRetrievedCFHTweets	Gets the list of Call For Help tweets.	Void	ArrayList<CallForHelpTweet> - list of Call For Help tweets	Public
setRetrievedCF	Sets the list of	ArrayList<CallFo	Void	Public



Method Name	Description	Parameters	Returns	Constraint
HTweets	Call For Help tweets.	rHelpTweet> - list of Call For Help tweets		
getRetrievedCA DTweets	Gets the list of Casualties and Damage tweets.	Void	ArrayList<CasualtiesAndDamage Tweet> - list of Casualties and Damage tweets	Public
setRetrievedCA DTweets	Sets the list of Casualties and Damage tweets.	ArrayList<CasualtiesAndDamage Tweet> - list of Casualties and Damage tweets	Void	Public
getRetrievedCA Tweets	Gets the list of Caution and Advice tweets.	Void	ArrayList<CautionAndAdviceTweet> - list of Caution and Advice tweets	Public
setRetrievedCATweets	Sets the list of Caution and Advice tweets.	ArrayList<CautionAndAdviceTweet> - list of Caution and Advice tweets	Void	Public
getRetrievedDTweets	Gets the list of Donation tweets.	Void	ArrayList<DonationTweet> - list of Donation tweets	Public
setRetrievedDTweets	Sets the list of Donation tweets.	ArrayList<DonationTweet> - list of Donation tweets	Void	Public

3.8.11 Binder Class

Description	The Binder Class maps the extracted information to its corresponding Ontology models that will be used to input into the ontology.
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
bindCA	Maps the Sentence object to the CautionAndAdvice	Sentence – CA labelled	CautionAndAdviceTweet – binded information	Public



Method Name	Description	Parameters	Returns	Constraint
	ceTweet			
bindCD	Maps the Sentence object to the CasualtiesAndDamageTweet	Sentence – CD labelled	CasualtiesAndDamageTweet – binded information	Public
bindCH	Maps the Sentence object to the CallForHelpTweet	Sentence – CH labelled	CallForHelpTweet – binded information	Public
bindD	Maps the Sentence object to the DonationTweet	Sentence – D labelled	DonationTweet – binded information	Public

3.9 Others

3.9.1 DBFactory Class

Description	The DBFactory Class handles the creation of connection to the database.
Type	Abstract

Attributes	Data Type	Constraint	Description	
dbDriver	String	Private	Driver of the database	
dbUrl	String	Private	URL of the database	
dbName	String	Private	Database name	
username	String	Private	Username of the database	
password	String	Private	Password of the database	
Method Name	Description	Parameters	Returns	Constraint
getConnection	Creates the connection	Void	Connection (JDBC) – provides the connection to the MySQL server	Abstract
getInstance	Creates a DBConnection instance	Void	DBConnection – DB Connection factory	Public
getDbDriver	Get the dbDriver value	Void	String – the database driver	Public
setDbDriver	Set the dbDriver value	String – the database driver	Void	Public
getUrl	Get the dbUrl value	Void	String – the URL of database	Public



Attributes	Data Type	Constraint	Description	
setUrl	Set the dbUrl value	String – the URL of database	Void	Public
getDbName	Get the dbName value	Void	String – the database name	Public
setDbName	Set the dbName value	String – the database name	Void	Public
getUsername	Get the username value	Void	String – username	Public
setUsername	Set the username value	String – username	Void	Public
getPassword	Get the password value	Void	String - password	Public
setPassword	Set the password value	String - password	Void	Public

3.9.2 DBConnection Class

Description	The DBConnection Class generates the connection to the database.
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
getConnection	Creates the connection to the server	Void	Connection – connection to the database	Public

3.9.3 Reader Class

Description	The Reader Class reads the CSV File
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
readCSVFile	Reads the CSV file that contain the tweet corpus	String – path the corpus	List<Sentence> - parsed CSV file	Public



3.9.4 XmlParser Class

Description	The XmlParser Class is responsible for reading and writing XML
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
saveXML	Save the results of the information extraction in a XML file.	List<Sentence> - processed sentences, String - output	Void	Public
saveXMLCA	Create a XML node for CA instance	CautionAndAdviceTweet – binded Caution and Advice	Element – written to the XML	Private
saveXMLCD	Create a XML node for CD instance	CasualtiesAndDamageTweet - binded Casualties and Damage	Element – written to the XML	Private
saveXMLCH	Create a XML node for CH instance	CallForHelpTweet – binded Call for Help	Element – written to the XML	Private
saveXMLD	Create a XML node for CD instance	DonationTweet – binded Donation	Element – written to the XML	Private

3.9.5 DocumentFrequency Class

Description	The DocumentFrequency Class provides the word and its frequency of a corpus.
Type	Class

Attributes	Data Type	Constraint	Description
documents	List<Sentence>	Private	Contains the tweet corpora
wordFrequency	HashMap<String,Integer>	Private	Contains the list of words and its corresponding frequency
documentCount	Int	Private	The number of instance in the corpus



Method Name	Description	Parameters	Returns	Constraint
getWordFrequency	Get the wordFrequency	Void	Map<String,Integer> - list of words and its frequency	Public
setWordFrequency	Set the wordFrequency	Map<String,Integer> - list of words and its frequency	Void	Public
getDocumentCount	Get the document count	Void	Int – number of corpus	Public
setDocumentCount	Set the document count	Int – number of corpus	Void	Public
getWordFrequency	Gives the count of the word	String - word	Int – number of occurrences in the corpus	Public
getListOfWords	Get the list of distinct words from the corpus	Void	List<String> - list of words	Public
readCorpus	Reads the CSV file that contains the corpus	String – path to the corpus	Void	Public
countWordsFrequency	Counts the frequency of the words in the corpus	Void	Void	Public

3.9.6 WeightScorer Class

Description	The WeightScorer Class measures the TF-IDF scores of the words
Type	Class

Attributes	Data Type	Constraint	Description
categoryDataset	DocumentFrequency	Private	Contains the list of word and its frequency of the category dataset. (Dataset that only contained 1 category)
documentDataset	DocumentFrequency	Private	Contains the list of words and its frequency of the whole dataset
Weights	Map<String,Double>	Private	List of words and its computed TF-IDF Scores



Attributes	Data Type	Constraint	Description
stopwords	List<String>	Private	List of stop words
stopWordPath	String	Private	Path to the list of stop words

Method Name	Description	Parameters	Returns	Constraint
loadStopWords	Load the list of stop words	Void	Void	Private
computeTF	Computes the term frequency	String – the word to be computed	Double – computed term frequency	Private
computeTFIDF	Computes the TF-IDF of a word	String – the word to be computed	Double – computed term frequency – inverted document frequency	Private
computeWeights	Computes the TFIDF scores of all the words in the list	Void	Map<String,Double> - list of words and its computed TFIDF score	Public
getTop	Get the words with the top N TF-IDF scores	String – savePath, int – top N	Void	Public
isValid	Checks if the word is valid	String - word	Boolean – checks if the word does not contain special characters, links, diacritics	Private
saveResults	Save the results to a text file	List<Map.Entry<String,Double>> - list of words, String - outputPath	Void	Public
saveResults	Save the results to a text file	String – outputPath	Void	Public
computeSize	Returns the size of the Weight attribute	Void	Int – number of distinct word in the corpus	Public

3.9.7 NGramModeller Class

Description	The NGramModeller Class produces the top highest frequency n-grams.
Type	Class



Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
CharNGram	counts the frequency of n-gram	Int – n-gram, double - topN, String – saveFile, String path - corpus	Void	Public
saveFile	Save the list to a file	List<Map.Entry<String,Integer>> - list of n-gram and its frequency, String – save path	Void	Private
getTop	Get the top N highest ngram frequency. Result is save to a file.	String – save file, int - top N	Void	Private

3.9.8 Filter Class

Description	The Filter Class is used to clean the words.
Type	Class

Attributes	Data Type	Constraint	Description

Method Name	Description	Parameters	Returns	Constraint
hasSpecialCharacters	Check if the word contains special characters	String - word	Boolean – check if the word contains special characters	Public
hasLinks	Check if the word is a link	String - word	Boolean – check if the word is a link	Public
hasDiacritics	Check if the word contains diacritics	String - word	Boolean – check if the word contains diacritic	Public
isNumeric	Check if the word is numeric	String – word	Boolean – check if the word is a number	Public
removeNonAlphaNumeric	Removes the special	String - word	String – removes the special	Public



Method Name	Description	Parameters	Returns	Constraint
	characters in the string		characters	