

Aufgabenblatt 8

Ausgabe: 21.1.2009

Abgabe: 27.01.2009

Prolog

Ziel dieses Übungsblatts ist es, an kleinen praktischen Beispielen, die Grundlagen der Text Mining Techniken zu verstehen.

(1) Text Mining mit RapidMiner

Vorgaben:

- In der `corpus.zip` unter URLHIER befinden sich die notwendigen Texte, die Sie für die Ausführung der folgenden Aufgabe brauchen.
- Downloaden und installieren Sie den RapidMiner 4.3 Community Edition (AGPL Lizenz) ¹(<http://rapid-i.com/content/view/26/84/lang,en/>).

Aufgaben:

1.1 TFIDF - Implementierung in Java (3 Punkte) Extrahiere Text von eine paar ausgewählten Wikipedia Artikeln oder anderen Sourcen. Implementieren eine *List < String >* in einem Java program, welche die Dokumente enthält. Trenne die Texte bei Leerzeichen (Tipp: StringTokenizer), um eine *List < String >* an Wörtern zu erhalten. Berechne eine *Map < String, Integer >* welche die Vorkommen jedes Wortes im Dokument zählt. Selektiere ein Dokument und kalkuliere TFIDF für jedes unterschiedliche Wort im Dokument; die Ausgabe des Programs ist als eine Paarliste (*word, TFIDF*) gerankt.

1.2 RapidMiner Beispiellernprozess (3 Punkte) Machen Sie sich vertraut mit dem RapidMiner. Probieren Sie den Entscheidungsbaumalgorithmus C4.5 von R.Quinlan (J4.8 ist die abgewandelte Implementierung des Algorithmus; unter WEKA-package des RapidMiners zu finden) mit vorgegebenem Datenset `golf.aml` aus. Dafür müssen Sie nach dem Start der RapidMiner GUI die Datei `01_DecisionTree.xml` die sich unter dem Pfad: `RapidMiner-4.3/<workspace>/sample/02_Learner` befindet, öffnen. Lesen Sie die Process-Info durch. Ohne die Prozessparameter zu verändern, starten Sie den DecisionTree-Prozess. Die graphische Darstellung zeigt Ihnen das gelernte Entscheidungsbaummodell. **Welche Pfade können aus dem Model extrahiert werden?** Schreiben Sie die extrahierten Pfade in natürlicher Sprache auf, sodass man die Aussagen nachvollziehen kann.

Wechseln Sie in den *Edit Mode* und editieren Sie die Attribute die in `golf.aml` eingebunden wurden. **Was würden Sie als das Klassenlabel bezeichnen?**

1.3 Eigenen Prozess mit RapidMiner starten: (9 Punkte) Unter der Verwendung des kleinen Korpus `corpus.zip` sollten Sie einen eigenen Entscheidungsbaum-basierten Lernprozess durchführen und diesen anhand der Visualisierung analysieren. Bevor Sie das Modell mit Hilfe von RapidMiner generieren können, müssen Sie die vorgegebene Trainingsmenge aus der `corpus.zip` entsprechend verarbeiten:

Preprocessing mit Feature Selection Schreiben Sie in JAVA oder in Sprache Ihrer Wahl eine Routine, die jeden Artikeltext aus jeder XML-Datei des Korpus als Stringvektor ² speichert. Versehen Sie jeden Stringvektor mit einem Klassenlabel: positiv, negativ oder neutral (jeweils im Dateinamen angeben.). Indizieren Sie den vorbereiteten Korpus unter der Verwendung von der *Term Frequency- Inverse Document Frequency*-Methode, indem Sie einen Featurevektor für die Artikel erstellen:

¹Die Registrierung vor dem Download ist nicht zwingend erforderlich http://sourceforge.net/project/downloading.php?groupname=yale&filename=rapidminer-4.3-community-windows.exe&use_mirror=ovh

²hilfreich kann hier die Lucene-Bibliothek sein aus <http://lucene.apache.org/java/docs/>

$$\text{TFIDF} = \text{TF}(w,D) * \text{IDF}(w,|D|)$$

$\text{TF}(w,D)$ bezeichnet die Häufigkeit des Auftretens von Wort w im Dokument D

$\text{DF}(w)$ gibt die Anzahl der Dokumente an in denen das Wort w vorkommt

$\text{IDF}(x,|D|)$ legt fest, wie gut das Wort w ein gegebenes Dokument D

von anderen Dokumenten abgrenzt

$|D|$ gibt die Anzahl der Dokumente an

Das Klassenlabel des Vektor trägt den Namen *Trend*. Wählen Sie die errechneten 10 häufigste und 10 wichtigste Wörter des Korpus aus und schreiben Sie Ihre Beispielartikel als Lerninstanzen in das entsprechende Format in `beispielartikel1.data` und `beispielartikel2.data`. Erstellen Sie die `beispielartikel.aml`, welche die `beispielartikel1.data` verwendet. Beispiel:

Featurevektor = {Prozent, Absatzzahl, Aktie, Buy, Studienergebnisse, Trend}

Artikelvektor (nach TF) = {5, 2, 1, 3, 7, positiv}

Achten Sie auf die korrekte Typdefinition der Attribute!(nominal, real, integer, etc.)

Learning Starten Sie nun einen neuen Learnerprozess indem Sie den ProcessWizard verwenden und DataInput + Decision Tree Learning auswählen. Wählen Sie den J4.8 Algorithmus (unter

NewOperator -> Learner -> Supervised -> WEKA -> Trees

zu finden). Lernen Sie das Model mit Ihrem Beispielsset. Verwenden Sie auch die `beispielartikel2.data` und vergleichen die gelernten Modelle.

Model Analysieren Sie das erstellte Lernmodel. **Ist der mit einfacher statistik errechnete positive, negative oder neutrale Trend gut nachvollziehbar? Wie werden die Entscheidungspfade in diesem Modell generiert?**