УДК 681.301

ЗАЙЦЕВ В.Г.,
ЛАН ЧУНЬЛИНЬ

ПРОЕКТИРОВАНИЕ И ВНЕДРЕНИЕ СИСТЕМЫ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ДОКУМЕНТОВ ДЛЯ КОНЕЧНОГО МНОЖЕСТВА ЯЗЫКОВ

В статье исследуется практические аспекты создания автоматической системы распознавания и классификация документов. Определяются основные методы автоматического распознавания, кодировки и классификации документов для конечного множества языков. Описывается предлагаемая архитектура автоматической системы классификации текстов для конечного множества языков.

The paper investigates the practical aspects of creating an automated system for automatic language recognition and classification of documents. The basic methods for automatic language identification and classification of documents on proposed. Describes the proposed architecture of the automatic text classification system for multilingual environment.

Введение

В современном мире, действует целый ряд систем классификации больших объемов текстовой информации, в основе которых лежат технологии компьютерной лингвистики и алгоритмов распознавания образов. В настоящее время существует большое количество текстовых документов на различных языках. Сегодня в Евросоюзе(ЕС) говорят на 23 официальных языках. Официальные языки ЕС – это английский, голландский, греческий, датский, испанский, итальянский, немецкий, португальский, финский, французский и шведский. Официальные тексты должны публиковаться на всех упомянутых языках. Все решения, принимаемые официальными органами ЕС, переводятся на все официальные языки, и граждане ЕС вправе обращаться в органы ЕС и получать ответ на свои запросы на любом из официальных языков. Официальными языками Организации Объединённых Наций(ООН) являются: английский, арабский, испанский, китайский, русский и французский. Обычные системы автоматической классификации текстов решают задачу автоматической классификации текстов на одном языке. Таким образом, необходимо решить задачу автоматической классификации текстов как для одного языка, так и для задач автоматической классификации текстов на разных языках.

Создание автоматической системы классификации текстов для конечного множества языков, можно разделить на две подсистемы. Первая подсистема служит для автоматического распознавания языка(Language Recognition) и идентификации(Language Identification). Вторая из подсистем – для автоматической классификации текстов, из первый подсистемы.

Первая подсистема: автоматическое распознавание и кодировка языка

Подсистема для автоматического распознавания языка состоит из двух частей. Блок автоматической идентификация зыка, извлечение текста из файла произвольного формата, определение кодировки текста. Блок автоматического распознавания языка, разделяет текст на абзацы, предложения, слова и определяет язык. Определение языка документа основано на принципе максимального распознавания слов каждого из языков. Алгоритм автоматической идентификации предложений языка использует тексты в кодировке UTF-8 или UTF-16 (двухбайтовые символы).

Подсистема автоматического распознавания распознает язык документа. Сейчас распознается больше 50 различных комбинаций кодировки и языка. Распознавание реализовано с использованием так называемой "N-Gram-Based Text Categorization" технологии. В статье[1] был предложен метод для определения языка и кодировки документа по его содержимому на основании статистик документов, для которых язык и кодировка известны заранее. Метод определяется частотой N-грамм(подстрок или сочетаний символов, длиной не более N) предполагает, что примерно 300 самых часто используемых N-грамм сильно зависят от языка. Алгоритм включается в нахождении частот N-грамм для всех тестовых документов, для которых

известен язык, а также для каждого документа, язык которого нуждается в определении. После этого среди всех тестовых документов находим тот, для которого расстояние от его N-граммной статистики до статистики тестируемого документа минимально. После того языком тестируемого документа считается язык найденного тестового документа. Расстояние между статистиками считается следующим образом: все N-граммы сортируются в порядке убывания частоты их появления, затем для каждой N-граммы вычисляется разница её позиций в отсортированном списке N-грамм тестового и тестируемого документов. Расстояние между статистиками определяется как сумма разностей позиций каждой N-граммы:

$$l = \sum_{i=1}^{300} \left| P_i - \tilde{P}_i \right| \qquad (1)$$

Где $P_i, \tilde{P}_i$ – позиции i-й N-граммы в тестовом и тестируемом документах соответственно. Значение N предлагается использовать равным 5(включительно). [2]

Распознавание работает хорошо на текстах в 500 байт и длиннее. Более короткие тексты могут распознаваться хуже.

Вторая подсистема: автоматическая классификация текстов

Классификация документов – одна из задач информатики, заключающаяся в отнесении документа к одной из нескольких категорий на основании содержания документа. Использует методы информационного поиска и машинного обучения. Следует отличать классификацию текстов от кластеризации, в последнем случае тексты также группируются по некоторым критериям, но заранее заданные категории отсутствуют.[3]

Постановка задачи классификации

Определим задачу формально. Пусть задано некоторое конечное множество категорий $C = \{c_1, ..., c_{|c|}\}$, конечное множество документов $D = \{d_1, ..., d_{|D|}\}$ и некоторая вначале неизвестная целевая функция *Ф*, которая для каждой пары <документ, категория> определяет, соответствуют ли они друг другу: $\Phi: D \times C \rightarrow \{0,1\}$. Задача состоит в том, чтобы найти максимально близкую к функции *Ф* функцию *Ф′*. Функцию *Ф′* называют классификатором. Машинное обучение основывается на начальном множестве документов $\Omega = \{d_1, ..., d_{|\Omega|}\} \subset D$. При этом, значение целевой функции *Ф* известно для каждой пары $< d_j, c_i > \in \Omega \times C$. Документы из *Ω* разделяют на два непересекающиеся множества:

"Учебное" множество $T_r = \{d_1, ..., d_{|T_r|}\}$ и множество документов, с помощью которой создается классификатор *Ф′*. *Ф′* обучается индуктивно, основываясь на замеченных характеристиках этих документов.

"Тестовое" множество $T_e = \{d_{|T_r+1|}, ..., d_{|\Omega|}\}$ и множество документов, на котором тестируется эффективность построенного классификатора. Каждый "тестовый" документ подается на вход классификатора *Ф′*, а затем сравнивается результат классификатора $\Phi'(d_j, c_i)$ с известным значением функции $\Phi(d_j, c_i)$. Классификатор считается тем эффективнее, чем чаще эти значения совпадают.

Документ $d \in \Omega$, называется положительным или отрицательным примером для категории *c*, если значение функции *Ф(d,c)* равно *1* или *0*, соответственно. [4]

Первичная обработка документов

Выбор веса признаков и уменьшение размерности.

В статье [5] приводится подробное исследование различных подходов к выбору весов признаков. Результаты экспериментов, описанных в этой статье, показывают, что одной из лучших формул вычисления весов является:

$$W_{ij} = TF_{ij} * IDF_i \qquad (2)$$

Где *TF* – частота термина в документе, *IDF* – обратная частота документа. Каждый документ – это просто набор слов (термов). Множество всех термов обозначим как *T*. Каждый терм $t_i \in T$ имеет вес w_{ij} по отношению к документу $d_j \in D$. Таким образом, каждый документ можно представить в виде вектора весов его термов $\overrightarrow{d_j} = < w_{ij}, ..., w_{|T|j} >$. Веса документов нормируют так, чтобы $0 \le w_{ij} \le 1$ для $\forall i, j : 0 \le i \le |T|, 0 \le j \le |D|$. Здесь TF_{ij} - это отношение числа термов t_i в документе d_i к общему числу термов в этом документе, а IDF_i – число, обратное количе-

ству документов, в котором встречается терм.

Нормализовать вес терма в документе можно следующим стандартным способом:

$$w_{ij} = \frac{TF_{ij} * IDF_i}{\sqrt{\sum_{s=1}^{|T|}(TF_{sj} * IDF_s)^2}} \quad (3)$$

После приведения всех слов документа к нормализованной форме, полученное пространство признаков имеет очень большую размерность. Эту размерность можно существенно уменьшить без ухудшения качества классификации, если исключить слова, слабо влияющие на результаты. [6]

Обычно из списка признаков удаляют так называемые "стоп-слова"(stopword) и из списка признаков можно удалить слишком редко встречающиеся слова.

Кроме того часто применяется методы выделения слов с использованием критерия информационного веса слова в категории (mutual information gain). Информационный вес слова в категории определяется по формуле:

$$MI(x_i, c) = \sum_{x_i \in \{0,1\}} \sum_{c \in \{0,1\}} P(x_i, c) \log \frac{P(x_i, c}{P(x_i)P(} \quad (4)$$

Здесь

$$P(x_i = 1) = 1 - P(x_i = 0) =$$

$$= \frac{\begin{array}{c}количество\ документов,\\ содержащих\ слово\ x_i\end{array}}{количество\ всех\ документов},$$

$$P(c = 1) = 1 - P(c = 0) =$$

$$= \frac{\begin{array}{c}количество\ документов,\\ принадлежащих\ категории\ c\end{array}}{количество\ всех\ документов}$$

$P(x_i, c)$ – вероятности совместного распределения слов и категории. Если предположить, что распределения слова x_i и категории c статистически независимы, то легко видеть $MI(x_i, c) = 0$. Если же между встречаемостью слова x_i и категории c имеется строгая логическая зависимость, то $MI(x_i, c)$ – максимально. Метод сокращения размерности на основе выделения наиболее информационно-значимых слов применяется, например, в работе [7].

Методы машинной классификации

Существует несколько наиболее известных методов классификации:

1. Метод Байеса.

Метод Байеса основан на анализе совместных распределений признаков документа и категорий [8]. Документу $D = < d_1, d_2, ..., d_n >$ сопоставляется наиболее вероятная апостериорная категория, определяемая по формуле:

$$c^* = \arg\max_{c \in C} P(c \mid x_1 = d_1, x_2 = d_2, ..., x_n = c \quad (5)$$

Апостериорная вероятность принадлежности документа некоторой категории вычисляется по формуле Байеса, связывающей априорную вероятность с апостериорной:

$$P(c \mid x_1 = d_1, x_2 = d_2, ..., x_n = d_n) =$$

$$= \frac{P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n \mid c) \cdot P(c)}{P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n)} \quad (6)$$

Подставляя (6) в (5), получаем:

$$c^* =$$

$$= \arg\max_{c \in C} \frac{P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n \mid c) \cdot P(c)}{P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n)} \quad (7)$$

Так как знаменатель не зависит от категории, его можно исключить:

$$c^* = \arg\max_{c \in C} P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n \mid \cdot P(c) \quad (8)$$

Условные вероятности $P(x_1 = d_1, x_2 = d_2, ..., x_n = d_n \mid c)$ можно вычислить в предположении условной независимости переменных $x_1, x_2, ..., x_n$. В этом случае, формула для определения наиболее вероятной категории будет выглядеть следующим образом:

$$c^* = \arg\max_{c \in C} P(c) \cdot \prod_{i=1..n} P(x_i = d_i \mid c) \quad (9)$$

Для множества обучающих документов вероятности $P(x_i = d_i \mid c)$ вычисляются по формуле [9,10,11]:

$$P(x_i = d_i \mid c) = \\ = \frac{|\{D \in Ex \mid c \in Rub(D) \wedge D_i = d_i\}| + 1}{|\{D \in Ex \mid c \in Rub(D)\}|} \quad (10)$$

2. Метод опорных векторов SVM (Support Vector Machines).

Метод опорных векторов [12,13,14] разработан на основе принципа структурной минимизации риска – одновременного контроля количества ошибок классификации на множестве для обучения и «степени обобщения» обнаруженных зависимостей.

Нахождение оптимальной плоскости разделения множеств методом SVM сводится к решению оптимизационной задачи с линейными ограничениями типа равенств и неравенств [12]:

$$L_D(\alpha) = \\ = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \to \max, \\ 0 \le \alpha_i \le C, \sum_{i=1}^{l} \alpha_i y_i = 0 \quad (11)$$

Здесь $K(x_i, x_j)$ – функция ядра SVM, которая в простейшем случае равна евклидову скалярному произведению векторов x_i и x_j. Для решения задачи (11) предложены эффективные методы решения [15,16].

Оценка качества классификации

С целью определения наилучшего метода были предложены следующие критерии: полнота; точность. Для оценки качества классификации используются метрики из информационного поиска:

Полнота (recall): отношение количества найденных документов из категории к общему количеству документов категории.

Точность (precision): Определяется как отношение числа релевантных документов, найденных ИПС, к общему числу документов.

$$recall = \frac{|D_{rel} \cap D_{retr}|}{|D_{rel}|} \quad (12)$$

$$precision = \frac{|D_{rel} \cap D_{retr}|}{|D_{retr}|} \quad (13)$$

где D_{rel} – это множество релевантных документов в базе, а D_{retr} – множество документов, найденных системой.[17]

В статье [7,8,10,11] сравниваются различные методы машинного обучения. Результаты показали, что метод SVM имеет преимущество перед другими методами машинного обучения.

Результаты работы

Программа распознавания языка и классификации текстов написана на Java. Система распознает следующие языки: русский, китайский, украинский, английский, немецкий, французский. В данной работе для классификации использован SVM метод. Было протестировано три группы из 5000 документов. В качестве результатов средняя полнота: 86.3%. средняя точность: 89.2%. Основными недостатками описанного алгоритма являются: неустойчивость определения языка малых текстовых документов – отдельные предложения на похожих языках распознаются неуверенно и с ошибками. Кроме того, системы автоматической классификации текстов показывают лучшие временные результаты на 64-битной аппаратной платформе(ширина регистров современных процессоров), чем на 32-битной аппаратной платформе, которая работает медленней.

Выводы

Выбранный метод и результаты работы подтверждают возможность создания эффективной системы автоматической классификации документов для конечного множества языков по критерию принадлежности к определенной области знания, используя современные средства вычислительной техники.

Список литературы

1. Cavnar, W. B. and J. M. Trenkle, “N-Gram-Based Text Categorization” In Proceedings of Third Annual Symposiumon Document Analysis and Information Retrieval,Las Vegas, NV, UNLV Publications/Reprographics, pp. 161-175, 11-13 April 1994

2. Сотник С.Л. Идентификация языка UNICODE-текста по N-граммам длиной до 4-х включительно (квадрограммам). Журнал “Математичне моделювання” №1,2(15) 2006, стр. 111-112, Днепродзержинск, издательство ДГТУ.
3. http://ru.wikipedia.org/wiki/Классификация_документов
4. Юрий Лифшиц. Курс "Алгоритмы для Интернета" РАН 2006.
5. Salton G, Buckley C. Term-Weighting Approaches in Automatic Text Retrieval. // Information Processing and Management, —1988 — p. 513-523.
6. Yang Y., Pedersen J. A comparative study on feature selection in text categorization. // In: Proc. of ICML-97, 14th International Conf. On machine Learning — Nashville, USA, 1997. — p. 412-420.
7. Dumais S., Platt J,, Heckerman D., Sahami M. Inductive learning algorithms and representations for text categorization. // In Proc. Int. Conf. on Inform. and Knowledge Manage., 1998. — p. 2-6.
8. Yang Y., Liu X. A re-examination of text categorization methods. // Proc. of Int. ACM Conference on Research and Development in Information Retrieval (SIGIR-99), 1999 — p. 42-49.
9. Joachims T. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. // Proceedings of ICML-97, 14th International Conference on Machine Learning. — 1996. — p. 2-13.
10. Yang Y. An Evaluation of Statistical Approaches to Text Categorization. / Journal of Information Retrieval, 1999 — V.1 — p. 67--88.
11. Joachims T. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. // Proceedings of ECML-98, 10th European Conference on Machine Learning — 1998. — p. 6-18.
12. Vapnik V. The Nature of Statistical Learning Theory. — Springer-Verlag — New York, 1995. — p. 123-167.
13. Burges C.J.C. A tutorial on support vector machines for pattern recognition. // Data Mining and Knowledge Discovery, 1998. — p. 955-974,
14. Вапник В.Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979. —с. 223-255.
15. Joachims T. Making Large-Scale SVM Learning Practical. Advances in Kernel Methods // Support Vector Learning, Burges C., Smola A. (ed.), — MIT-Press, 1999. —p. 5-12.
16. Joachims T. Estimating the Generalization Performance of a SVM Efficiently. // Proceedings of the International Conference on Machine Learning, — Morgan Kaufman, 2000. —p. 5-22.
17. http://ru.wikipedia.org/wiki/Информационный_поиск

Поступила в редакцию 16.12.2009