Facharbeit zum Thema

Automatische Suche von sprachlich optimierbaren Wikipedia Artikeln mittels Analyse der formalen Gestaltungs- und sprachlichen Ausdrucksmerkmalen

Autor: Joseph Adams

Schule: Maria-Sybilla-Merian Gymnasium Krefeld

Kurs: IF-GK2 POK

Schuljahr: 2013/14

Inhaltsverzeichnis

1. Zielsetzung des Systems

Wikipedia ist momentan die größte Onlineenzyklopädie der Welt[1] und um die Qualität der Wikipedia beizubehalten, arbeiten jeden Tag viele freiwillige Autoren daran, Inhalte zu verbessern, Artikel zu erweitern und neue Artikel zu schaffen. Dieser Prozess des Verbesserns von schon existierenden Artikeln lässt sich wiederum in zwei Schritte unterteilen. Der erste Schritt ist das eigentliche Identifizieren von Mängeln in Artikeln und das damit verbundene Markieren bzw. Hinweisen auf diese Mängel. Der zweite Schritt ist das Korrigieren der zuvor gefundenen Mängeln.

Die angestrebte Aufgabe des im Folgenden beschriebenen Systems ist es, diesen ersten Schritt zu übernehmen, also Mängel in Artikeln zu identifizieren und ggf. zu markieren. Dies hat den Sinn, dass wenn diese Aufgabe von einer Maschine übernommen werden kann, die freiwilligen Autoren mehr Zeit zur Verfügung haben, die noch übrigen Aufgaben zu bearbeiten. Man spart Zeit.

Jedoch muss dies weiter eingeschränkt werden, denn dieses System kann nur Mängel erkennen, die die Struktur und sprachliche Gestaltung der Artikel betreffen, also keine inhaltlichen Fehler. Weiterhin definieren wir einen Mangel in einem Artikel als eine unzureichende Ähnlichkeit zu der Struktur von featured articles[2] (im strukturellen Kontext) oder einem verwendeten Wortschatz, der nicht der Gesamtheit aller anderen Wikipedia Artikel ähnelt (im lexikalischen Kontext).

2. Funktionsweise des Systems

2.1. Definition der Arten

Wie bereits zuvor genannt, überprüft das System die Ähnlichkeit von einem Artikel zu den featured articles bzw. zu allen anderen Artikeln. Um dies

---

[1] “Wikipedia - Wikipedia, the free encyclopedia” [online], update 12. Januar 2014, http://en.wikipedia.org/wiki/Wikipedia, 12.01.14

[2] siehe Anmerkungsapparat

durchführen zu können, bedarf es eines Satzes an Merkmalen (von hieran auch features genannt), anhand derer man Ähnlichkeit feststellt. Für die Analyse der formalen Gestaltung (Struktur) werden die Anzahl der Abschnitte, Links[3], Fußnoten, Sätze, Wörter, die Link-Dichte[4] und durchschnittliche Satzlänge verwendet (dies ist natürlich um weitere Merkmale erweiterbar). Von der Verwendung der Wortlänge als Feature wurde abgesehen, da es sich gezeigt hat, dass diese im Mittel immer bei 5,1 Buchstaben je Wort lag.

Die Merkmale um den sprachlichen Ausdruck zu analysieren sind die einfachen Wörter und ihre dazugehörigen Bi- und Tri-Gramme[5].

2.2. Berechnung der Merkmale

Zur Berechnung der Merkmale werden die Artikel von der Dump-Datei[6] einzeln gelesen und in einem Puffer abgelegt, sodass bei Bedarf sofort (bei entsprechender Puffergröße) ein weiterer Artikel zur Verfügung steht. Sobald ein Artikel aus dem Puffer geholt wird, sendet man ihn durch eine Bearbeitungspipeline, in dem der Artikel zunächst markiert wird, um aufzuzeigen, welcher Kategorie er später zugeordnet wird: featured articles werden von “normalen” Artikeln getrennt und Weiterleitungsartikel[7] werden gelöscht, da diese für das System irrelevant sind. Darauf wird der Artikel von etwaigem Wiki-Markup gesäubert[8], sodass nur noch die Klarschrift vorhanden ist. Dies macht die weitere Verarbeitung leichter. Nun werden die eigentlichen Merkmale berechnet bzw. die N-Gramme gebildet. Dies geschieht durch modulare Funktionen[9], die sich auch in der Pipeline befinden. Es können sich zu jeder Zeit n verschiedene Artikel in der

---

[3] es werden hier interne von externen Links unabhängig betrachtet, hierfür fallen also 2 features an

[4] berechnet sich aus: (interne links + externe links) / Anzahl der Wörter

[5] siehe Anmerkungsapparat

[6] siehe Anmerkungsapparat

[7] siehe: “Wikipedia:Redirect - Wikipedia, the free encyclopedia” [online], update 10. Januar 2014, http://en.wikipedia.org/wiki/Wikipedia:Redirect, 12.01.14

[8] Geschieht mittels regulären Sprachen. Komplexität wächst O(n) mit der Artikellänge

[9] Es ist wichtig zu nennen, dass diese Funktionen komplett von einander unabhängig sind.

Pipeline befinden, wo n die Anzahl der modularen Funktionen der Pipeline ist. Dadurch, dass die einzelnen Bearbeitungsschritte klar voneinander getrennt sind, lassen sich einfach neue hinzufügen bzw. ungewollte Merkmale leicht entfernen. Weiterhin lässt diese Bauweise es zu, dass die Berechnung von Merkmalen, in verschiedenen Pipelines, parallel[10] ablaufen kann. Hierbei werden mehrere Pipelines durch die Artikel-Quelle (Puffer, dessen Größe natürlich entsprechend der Anzahl der Pipelines erhöht werden muss) gespeist und vor der Spaltung in Kategorien wieder mit Hilfe des Fan-In Prinzips zusammengeführt[11]. In aktueller Implementation werden 4 Pipelines gleichzeitig gespeist, mit denen es möglich war alle features aller Wikipedia Artikel innerhalb von 72 Stunden zu berechnen und für die Weiterverwendung (siehe 2.5) abzuspeichern. Prinzipiell ist es möglich die Anzahl der Pipelines weiter zu erhöhen, doch gab es bis jetzt noch keinen Anlass für diese Maßnahme, da das Go concurrency Modell[12] das Hinzufügen weiterer Prozessoren/Prozessor-Kernen optimal ausnutzen kann, indem es die modularen Funktionen in der Pipeline, welche in sog. Goroutines ausgeführt werden, auf alle Prozessoren/Prozessor verteilen kann und somit die Ausführung der Pipeline automatisch schneller geschieht.

2.3. Trainieren des neuronalen Netzwerks

Nachdem nun die Merkmale der Artikel berechnet wurden, kann das neuronale Netzwerk trainiert werden, welches dafür zuständig ist, die Ähnlichkeit der formalen Gestaltungsmerkmale der non-featured articles zu den featured articles festzustellen.

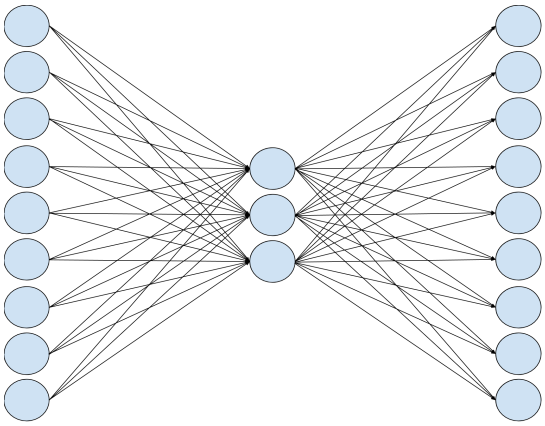
---

[10] eigentlich ist nur Nebenläufigkeit garantiert

[11] Siehe Anhang für ein Beispiel des Fan-In Prinzips. Modelliert nach Vorbild: Rob Pike, “Google I/O 2012 - Go Concurrency Patterns”, Juni 2012, http://www.youtube.com/watch?v=f6kdp27TYZs [online]

[12] siehe Anmerkungsapparat für Erläuterung

Das System verwendet ein feed-forward neuronales Netzwerk[13], bei dem die Daten der Neuronen immer nur in eine Richtung (der Ausgabeschicht) weitergegeben werden. Im Falle dieses Systems haben wir die simpelste Variante eines feed-forward neuronalen Netzwerks gewählt, bei dem alle Neuronen einer Schicht mit allen Neuronen der nächsten Schicht verknüpft sind (siehe Schaubild zur Verdeutlichung).



Dieses Netzwerk wird mittels Fehlerrückführung[14] und einer nicht-linearen Variante des CG-Verfahrens[15] trainiert. Der Minimierer versucht hierbei ein lokales Minimum der nicht-linearen Kostenfunktion[16] zu finden.

Hier stellt sich nun jedoch ein Problem, nämlich, dass wir nur featured articles (also positive Beispiele von Artikeln) zum Trainieren des neuronalen Netzwerks haben, was normalerweise zu einem "Übersättigungsproblem" führt, welches das neuronale Netzwerk nutzlos macht. Wir lösen dieses Problem mit Hilfe der Technik, die in *"Document Classification on Neural Networks Using Only Positive Examples"*[17] beschrieben wird. Dort heißt es:

> *The basic design of the filter under discussion here is a feed-forward neural network. In order to incorporate the restriction of positive examples only, we*

---

[13] Für weiter Informationen siehe: "Feedforward neural network - Wikipedia, the free encyclopedia" [online], update 9. August 2013, http://en.wikipedia.org/wiki/Feedforward_neural_network, 26.01.14

[14] siehe: "Backpropagation – Wikipedia" [online], update 9. September 2013, http://de.wikipedia.org/wiki/Backpropagation, 26.01.14

[15] siehe: "Nonlinear conjugate gradient method - Wikipedia, the free encyclopedia" [online], update 22. März 2013, http://en.wikipedia.org/wiki/Nonlinear_conjugate_gradient_method, 26.01.14

[16] siehe Anhang für weitere Informationen über die Kostenfunktion

[17] Larry M. Manevitz & Malik Yousef, "Document Classification on Neural Networks Using Only Positive Examples", Department of Computer Science, University of Haifa, Haifa, Israel, 2000

used the design of a feed-forward network with a "bottleneck", we choose a three level network with m inputs, m outputs and k neurons on the hidden level, where k < m. Then the network is trained, under standard back-propagation to learn the identity function on the sample examples.

The idea is that while the bottleneck prevents learning the full identity function on m-space; the identity on the small set of examples is in fact learnable. Then the set of vectors for which the network acts as the identity function is a sort of sub-space which is similar to the trained set. (This avoids the "saturation" problem of learning from only positive examples.) Thus the filter is defined by applying the network to a given vector; if the result is the identity, then the vector is "interesting".

Es hat sich im Laufe der Implementation gezeigt, dass sich diese Idee auch auf das Finden von ähnlichen Artikeln übertragen lässt. Somit können wir uns dies auch zu Nutze machen, indem wir das neuronale Netzwerk darauf trainieren die Identitätsfunktion der Eingabe (in diesem Fall der Feature-Vektor eines featured articles) zurück zu geben. Weiterhin bauen wir, wie beschrieben, auch einen Engpass in der versteckten Schicht des neuronalen Netzwerks ein[18], der eine Art verlustbehaftete Kompression verursacht und somit verhindert, dass die Identitätsfunktion für alle Vektoren in R^9 (für genaue Auflistung der features, siehe: 2.1) gelernt wird. Ebenfalls hat sich gezeigt, dass das neuronale Netzwerk schneller trainiert werden kann und bessere Ergebnisse produziert, wenn die Parameter im Feature-Vektor $0 < x_i < 1$ sind, wobei x_i ein feature in einem (allgemeinen) Feature-Vektor ist. Dies liegt wahrscheinlich daran, dass wenn die Parameter in den Feature-Vektoren alle zwischen 0 und 1 sind, die Kostenfunktion des neuronalen Netzwerks nicht so verzerrt ist und der Minimierer schneller konvergiert. Allgemeines Bsp.: $x_{neu} = (\begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T - min) \circ (min - max)$ wobei min ein Vektor mit dem für

[18] In aktueller Implementation ist die Anzahl der Neuronen in der versteckten Schicht des Netzwerks gleich ceil(0.3*Anzahl der features)

ein jedes feature minimal registrierten Wert in den featured articles und max mit den entsprechenden maximalen Werten in den featured articles.

2.3.1. Aufstellen der Schwellenwert-Matrix

Da das neuronale Netzwerk nicht genau die Identitätsfunktion abbilden kann, ist es hilfreich, wenn man eine Schwellenwert-Matrix berechnet, die jemandem einen guten Überblick über die Abweichung von der (echten) Identitätsfunktion geben kann. Auch wird diese in 2.5.1 dazu verwendet die finale Analyse von Artikeln durchzuführen.

Man berechnet sie, indem man den Vorhersagefehler aller features für alle featured articles (bzw. des Trainingssets) errechnet und dann für jedes Feature separat die durchschnittliche Abweichung mit der Standardabweichung der Abweichung (zu der echten Identitätsfunktion) addiert. Dabei ist der Vorhersagefehler: $predictionErr_i = (x_i - y_i)^2$ des i-ten featured articles. Für ein simples Rechenbeispiel siehe Anhang.

2.4. Zusammenstellung der N-Gramme & Good-Turing smoothing

Wie bereits in Abschnitt 2.2 erwähnt ist die Extraktion der N-Gramme auch ein Teil der Bearbeitungspipeline, jedoch müssen die N-Gramme nach Durchlaufen der Pipeline von den Artikeln getrennt werden und in einem einzigen Modell (Gesamt-Modell) zusammengeführt werden. Dies gestaltet sich relativ einfach, da die N-Gramme, sowohl in den einzelnen Artikeln als auch in dem Gesamt-Modell, in einer Map vorliegen und diese nur vereinigt werden müssen. Erst nach diesem Prozess, also nach Durchlaufen aller Artikel, die man in seiner Analyse berücksichtigen möchte (zumeist alle), kann damit begonnen werden die angepasste Good-Turing Wahrscheinlichkeit zu berechnen. Dies geschieht wie folgt, für bereits gesehene Wörter ergibt sich eine Wahrscheinlichkeit von:

$$P_{GT}(X) = \frac{N_X}{T} \cdot (1 - \frac{E(1)}{T})$$

und für noch nicht gesehene Wörter ergibt sich:

$$P_{GT}(neu) = \frac{E(1)}{T}$$

Dabei ist $P_{GT}(X)$ die angepasste Wahrscheinlichkeit das Wort X anzutreffen, bei $P_{GT}(neu)$ steht neu für ein unbekanntes Wort, T sei die Anzahl der angetroffenen Worte, N_X sei die Häufigkeit des Wortes X im gesamten Korpus und $E(n)$ sei die Häufigkeit ein Wort n-mal anzutreffen. Eine nähere Beschreibung dieses Verfahrens ist zu finden in: “Lecture 4 - Good-Turing probability estimation.”[19].

Streng genommen ist diese Anpassung der Wort Häufigkeiten und damit auch Wahrscheinlichkeiten nicht notwendig, jedoch erschien sie aus zwei Gründen sinnvoll. Zum einen gehen wir davon aus, dass sich das Vokabular eines Artikels, nachdem er einer Änderung unterzogen wurde nicht so drastisch ändert, als dass es eine signifikante Auswirkung auf die gesamten Häufigkeiten hätte und man diesen Schritt der Analyse, also das Aufstellen der N-Gramme, bei kurzzeitig aufeinanderfolgenden Analysen aller Artikel auslassen kann. Bei diesen Analysen könnte man dann auf das schon bestehende N-Gramm Modell zurückgreifen, die Analyse somit drastisch beschleunigen und vermeiden, sollten neue/unbekannte Worte auftreten, dass die Satz-Wahrscheinlichkeit (Definition folgt) auf 0 absinkt. Der zweite Grund ist die Simulation der Realität. Da man davon ausgehen kann, dass sich nicht alle Wörter einer Sprache in allen Wikipedia Artikeln (bzw. denen die man analysiert) finden lassen, sorgt man so dafür, dass der Mangel dieser fehlenden Wörter ausgeglichen wird.

Mit diesen angepassten Wort-Wahrscheinlichkeiten (von hier an auch generalisiert auf N-Gramme) ist es uns nun möglich die Analyse des sprachlichen Ausdrucks zu beschreiben. Diese beschränkt sich auf einen Vergleich der gesamten (Sprach-)Wahrscheinlichkeit eines Artikels mit der durchschnittlichen (Sprach-)Wahrscheinlichkeit aller Artikel. Sollte die (Sprach-)Wahrscheinlichkeit eines Artikels weniger als 90% der

[19] Greg Kochanski, “Lecture 4 - Good-Turing probability estimation.”, University of Oxford, 6. März 2006

durchschnittlichen (Sprach-)Wahrscheinlichkeit aller Artikel sein gilt (für uns) sein sprachlicher Ausdruck als unzulässig. Formal heißt dies, wenn: $P(A_n) < P_A$ gilt, ist der sprachliche Ausdruck eines Artikels unzulässig. Hierbei sei P_A die durchschnittliche (Sprach-)Wahrscheinlichkeit aller Artikel:

$$\frac{1}{N} \sum_{k=1}^{N} P(A_k)$$

dabei sei N die Anzahl der Artikel, A_n der n-te Artikel und $P(A_n)$ dessen (Sprach-)Wahrscheinlichkeit:

$$\left[\prod_{k=1}^{S_3} P_{GT}\left(a_{3,k}\right)\right] \cdot \left[\prod_{l=1}^{S_2} P_{GT}\left(a_{2,l}\right)\right] \cdot \left[\prod_{m=1}^{S_1} P_{GT}\left(a_{1,m}\right)\right]$$

wobei wiederum S_n die Anzahl der n-gramme des zu betrachtenden Artikels sind, P_{GT} die zuvor definierte (Good-Turing) andgepasste Wahrscheinlichkeit eines Wortes bzw. N-Gramms und $a_{x,y}$ das y-te x-Gramm ist. (In diesem Term finden sich drei Faktoren, da das System die Uni-, Bi- und Tri-Gramme betrachtet.)[20]

Realisiert werden die Berechnungen dieser Wahrscheinlichkeiten mittels logarithmischen Wahrscheinlichkeiten, um einem bestehenden arithmetischen Unterlauf zu entgehen und die Berechnungen zu beschleunigen, da bei dem Produkt zweier Wahrscheinlichkeiten x und y (laut Logarithmengesetzen) gilt: $log(x \cdot y) = log(x) + log(y)$

und die Summe zweier Zahlen schneller berechnet werden kann als das Produkt.[21] [22]

---

[20] Hierbei half zur Orientierung: Kathleen McKeown, “CS4705 - N-Grams and Corpus Linguistics”, University of Columbia, Datum unbekannt

[21] Für weitere Informationen siehe: “Log probability - Wikipedia, the free encyclopedia” [online], update 2. April 2013, http://en.wikipedia.org/wiki/Log_probability, 10.02.14

[22] Außerdem ist es nicht zwangsläufig nötig die logarithmische Wahrscheinlichkeit wieder in eine “normale” Wahrscheinlichkeit umzurechnen.

2.5. Analyse der Artikel

Nun da alle, für die eigentliche Analyse nötigen, Teile des Systems beschrieben wurden, können die Artikel analysiert werden. Wir unterscheiden bei der Analyse der Struktur eines Artikels zwischen zwei verschiedenen Anwendungsfällen: mit und ohne Schwellenwert-Matrix. Die Vorbereitung des Vektors, welcher die Abweichung der berechneten Identitätsfunktion zur eigentlichen Identitätsfunktion beinhaltet, ist in beiden Fällen die selbe. Der Feature-Vektor eines zu analysierenden Artikels wird geladen und als Eingabe in das neuronale Netzwerk gegeben. Darauf wird mit der Ausgabe des Neuronalen-Netzwerks der Vorhersagefehler (definiert in 2.3.1) der Identitätsfunktion des Feature-Vektors dieses Artikels berechnet. Dieser Vorhersagefehler ist der Ausgangspunkt der zwei verschiedenen Anwendungsfälle. (Der Analysevorgang für die sprachliche Gestaltung eines Artikels wurde bereits in 2.4 dargelegt)

2.5.1. Mit Schwellenwert-Matrix

Die Analyse mit Hilfe der in 2.3.1 aufgestellten Schwellenwert-Matrix ist dafür gedacht, dass sich ein freiwilliger Autor einen Überblick über die Mängel eines beliebigen Artikels machen kann. Hierbei wird aus dem Vorhersagefehler-Vektor ein Binärwert-Vektor ermittelt, welcher für jedes feature, sollte ein Mangel bestehen, eine “1” enthält und sonst eine “0”. Ein Beispiel hierfür wäre: $[1\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 1]^T$. Dieser Vektor zeigt an, dass der dazugehörige Artikel Mängel aufweist in: Durchschnittlicher Satzlänge, Anzahl externer Links, Anzahl der Abschnitte, Anzahl der Zitate und genereller Länge (Anzeichen hierfür sind, dass die beiden letzten Einträge des Vektors “1” sind). Die simple Form des Vektors (nur zwei mögliche Werte), erlaubt es uns jeden Eintrag (des Vektors) mit einem, auf Mängel hinweisendes, “template”[23] in Verbindung zu bringen, wie z.B.

[23] Genauer: mit “Wikipedia message box templates - Wikipedia, the free encyclopedia” [online], update 23. März 2013, http://en.wikipedia.org/wiki/Category:Wikipedia_message_box_templates, 15.02.14

“Template:External links” oder “Template:Refimprove”. Dies ermöglicht es auch freiwilligen Autoren aufgedeckte Mängel zu finden, ohne mit dem eigentlichen System interagieren zu müssen.

2.5.2. Ohne Schwellenwert-Matrix

Der zweite Anwendungsfall bezieht sich auf Artikel die als Kandidaten für featured articles oder auch “Good article[s]”[24] aufgenommen wurden. Es wäre möglich nur mittels des Vorhersagefehler-Vektors zu erkennen, welche die schwerwiegendsten Mängel eines solchen Kandidaten wären, um diese priorisiert zu beheben und somit eine Aufnahme in die featured articles oder “Good articles” zu beschleunigen. (Diese priorisierte Behebung von Mängeln ließe sich natürlich auch auf alle anderen Artikel übertragen, doch schien dort der Nutzen geringer.)

3. Verbesserungsmöglichkeiten

Nach erstmaliger Analyse aller Wikipedia Artikel fanden wir, dass das System an einigen Stellen verbesserungswürdig ist. Im folgenden möchten wir drei dieser Probleme beleuchten und mögliche Lösungen vorstellen. Zum einen wurde uns im Laufe der Zeit klar, dass die neun Merkmale (siehe 2.1), die das neuronale Netzwerk untersucht nicht ausreichen um einen Artikel vollkommen zu beschreiben. Das Hinzufügen von weiteren Merkmalen, die analysiert werden, liegt als Lösung dieses Problems nahe. Außerdem könnte man zusätzlich noch einen Vergleich eines Artikels mit anderen Artikeln seiner Kategorie hinzuziehen um eventuelle fachliche Abweichungen in der Struktur der Artikel auszugleichen (z.B. ein Artikel über Fotografie wird mehr Bilder enthalten als ein Artikel über Mathematik). Weitehin stellte es sich als sehr ungünstig heraus nur die Fähigkeit zu besitzen Artikel sequenziell (von der Dump-Datei) lesen zu können. Lösung hierfür wäre z.B. das Laden der Artikel in eine Datenbank, welche Direktzugriff auf alle beliebigen Artikel ermöglicht. Überdies haben wir

[24] “Wikipedia:Good articles - Wikipedia, the free encyclopedia” [online], update 15. August 2013, http://en.wikipedia.org/wiki/Wikipedia:Good_articles, 15.02.14

festgestellt, dass das neuronale Netzwerk an einer Überanpassung an die featured articles leidet. Durch das Erhöhen des Regularisierungsparameters (i.Mo. $\lambda = 0$) könnte dieses Problem behoben werden, allerdings wäre es auch möglich das neuronale Netzwerk nicht nur mit featured articles zu trainieren, sondern auch mit "Good articles", von denen es eine Vielzahl mehr gibt[25].

4. Schlussfolgerungen

Wikipedia wächst jeden Tag um neue Inhalte und es werden immer wieder alte verbessert. Jedoch müssen wir davon ausgehen, dass Menschen alleine, bzw. die Anzahl der freiwilligen Autoren, auf Dauer nicht ausreichen werden, um die aktuell vorherrschende Qualität der Wikipedia beizubehalten, bzw. diese noch weiter zu steigern. Dieses System bietet durch seine Autonomie einen ersten Schritt in Richtung Entlastung der Menschen (freiwilligen Autoren), welche Langfristig nötig werden wird um die Wikipedia zu erhalten.

5. Anmerkungsapparat

featured articles	Featured articles sind Wikipedia Artikel, welche die "aktiven Mitarbeiter[n] am Wikipedia-Projekt"[26] als äußerst lesenswert eingestuft haben. Aktuell sind 4,171 Artikel als "featured" eingestuft (Stand: 15.02.14).
N-Gramme	N-Gramme sind Textfragmente, die aus einem Wort/einer Silbe (Uni-Gram) bzw. mehreren Worten/Silben (Bi-, Tri-, Quad-Gramme, etc.) bestehen.

[25] Verhältnis: 4171:19397 (featured articles zu "Good articles"), stand 15.02.14
[26] "Wikipedia:Wikipedianer - Wikipedia" [online], update: 1. Dez 2013, http://de.wikipedia.org/wiki/Wikipedia:Wikipedianer, 06.01.14

Dump-Datei	Die Dump-Datei ist eine komprimierte Datei, welche alle Wikipedia Artikel (einer bestimmten Sprache) enthält. Sie wird jeden Monat aktualisiert. Die aktuelle (englische) Version ist hier erhältlich: http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2
Go concurrency Modell	Das Go (http://golang.org) concurrency Modell ist der Mechanismus mit dem Go Nebenläufigkeit in Programmen ermöglicht. Es basiert auf günstigen "goroutines" die nebenläufig agieren (somit Parallelität ermöglichen) und über sog. "channels" kommunizieren. Das Modell wurde inspiriert durch "Communicating sequential processes"[27], welches sich auch in einer anderen von Rob Pike mitentwickelten Sprache (Limbo) wiederfindet.

6. Anhang

Quelltext des Systems

siehe: https://github.com/jcla1/wikipedia_analyser/

Fan-in Beispiel

siehe: http://play.golang.org/p/N3BrGAjt8o

Kostenfunktion des neuronalen Netzwerks

Um das neuronale Netzwerk zu trainieren braucht man eine Kostenfunktion, welche die "Kosten" angibt, die entstehen wenn man einen

[27] "Communicating sequential processes - Wikipedia, the free encyclopedia" [online], update 12. Februar 2014, http://en.wikipedia.org/wiki/Communicating_sequential_processes, 18.02.14

bestimmten Satz an Übergangsmatrizen Θ benutzt. Wir gebrauchen eine logarithmische Kostenfunktion mit Regularisierung:

$$J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \cdot log\left(\left(h_\Theta(x^{(i)})\right)_k\right) + \left(1 - y_k^{(i)}\right) \cdot log\left(\left(h_\Theta(x^{(i)})\right)_k\right)\right] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(\Theta_{j,i}^{(l)}\right)^2$$

Hierbei ist $h_\Theta(x^{(i)})_k$ die k-te Ausgabe des neuronalen Netzwerks (mit Parametern Θ), x ist eine Matrix mit den Trainingsbeispielen (hier: die featured articles) und y ist eine Matrix mit den erwarteten Ausgaben des neuronalen Netzwerks (hier: $x = y$, da wir die Identitätsfunktion lernen möchten). Die Begrenzungen der Summen sind die Anzahl der Ein- bzw. Ausgaben des neuronalen Netzwerks und die Dimensionen der Übergangsmatrizen bzw. des neuronalen Netzwerks.

Schwellenmatrix Rechenbeispiel

Seien X und Y die Matrizen mit jeweils den Feature-Vektoren der featured articles und der Ausgabe des neuronalen Netwerks bei Eingabe des jeweiligen Feature-Vektor.

$$X = [x_1^T\ x_2^T\ x_3^T]^T \qquad Y = [y_1^T\ y_2^T\ y_3^T]^T$$

Dann berechnet man den Durchschnitt des Vorhersagefehlers (siehe 2.3.1)

$$means_i = \overline{(X_i - Y_i)^2}$$

Und die Standardabweichung des Vorhersagefehlers, wobei n die Anzahl der featured articles ist (Anzahl der Reihen in X):

$$stdDev_i = \sqrt{\frac{\left((X_i - Y_i)^2 - means_i\right)^2}{n}}$$

Wenn man diese nun addiert erhält man die Schwellenmatrix bzw. eigentlich ist es ein Vektor $\in R^n$:

$$s = mean + stdDev$$

7. Literaturverzeichnis

- Kathleen McKeown, “CS4705 - N-Grams and Corpus Linguistics”, University of Columbia, Datum unbekannt
- Larry M. Manevitz & Malik Yousef, “Document Classification on Neural Networks Using Only Positive Examples”, Department of Computer Science, University of Haifa, Haifa, Israel, 2000
- Greg Kochanski, “Lecture 4 - Good-Turing probability estimation.”, University of Oxford, 6. März 2006
- Rob Pike, “Google I/O 2012 - Go Concurrency Patterns”, Juni 2012, http://www.youtube.com/watch?v=f6kdp27TYZs [online]
- “Backpropagation – Wikipedia” [online], update 9. September 2013, http://de.wikipedia.org/wiki/Backpropagation, 26.01.14
- “Communicating sequential processes - Wikipedia, the free encyclopedia” [online], update 12. Februar 2014, http://en.wikipedia.org/wiki/Communicating_sequential_processes, 18.02.14
- “Feedforward neural network - Wikipedia, the free encyclopedia” [online], update 9. August 2013, http://en.wikipedia.org/wiki/Feedforward_neural_network, 26.01.14
- “File:Artificial neural network.svg - Wikipedia, the free encyclopedia” [online], update 22. Februar 2011, http://en.wikipedia.org/wiki/File:Artificial_neural_network.svg, 15.02.14
- “Log probability - Wikipedia, the free encyclopedia” [online], update 2. April 2013, http://en.wikipedia.org/wiki/Log_probability, 10.02.14
- “Nonlinear conjugate gradient method - Wikipedia, the free encyclopedia” [online], update 22. März 2013, http://en.wikipedia.org/wiki/Nonlinear_conjugate_gradient_method, 26.01.14
- “Wikipedia:Featured articles - Wikipedia, the free encyclopedia” [online], update 18. Februar 2014 http://en.wikipedia.org/wiki/Wikipedia:Featured_articles, 18.02.14
- “Wikipedia:Good articles - Wikipedia, the free encyclopedia” [online], update 15. August 2013, http://en.wikipedia.org/wiki/Wikipedia:Good_articles, 15.02.14
- “Wikipedia message box templates - Wikipedia, the free encyclopedia” [online], update 23. März 2013, http://en.wikipedia.org/wiki/Category:Wikipedia_message_box_templates, 15.02.14
- “Wikipedia:Redirect - Wikipedia, the free encyclopedia” [online], update 10. Januar 2014, http://en.wikipedia.org/wiki/Wikipedia:Redirect, 12.01.14
- “Wikipedia - Wikipedia, the free encyclopedia” t, update 12. Januar 2014, http://en.wikipedia.org/wiki/Wikipedia, 12.01.14
- “Wikipedia:Wikipedianer - Wikipedia” [online], update: 1. Dez 2013, http://de.wikipedia.org/wiki/Wikipedia:Wikipedianer, 06.01.14

8. Erklärung der Eigenständigkeit

Ich erkläre, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe.

Ort, Datum, Unterschrift