Joseph Birkner
Mat.Nr. 11393024
Protokolle zum Colloquium (03.07.2017)

Student: Pascal Guldener

„Neural Networks for Named Entity Recognition“

Dieses Referat hat Pascal ursprünglich für seine Seminararbeit bei Alexander Phraser ausgearbeitet. Es basiert auf dem Paper „Natural Language Processing (Almost) from Scratch“, veröffentlicht 2011 von Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu und Pavel Kuksa (Collobert et al.). Der Titel bezieht sich auf die Motivation des des Papers, ein „one-fits-all“ Modell zur Lösung mehrerer klassicher computerlinguistischer Probleme gleichzeitig zu entwickeln, ohne dabei auf etablierte Systeme, oder gar etablierte linguistische Klassifikationsmerkmale, zurückzugreifen.

Das enstandene System wurde in C implementiert, und trainiert für Named Entity Recognition (NER), Part-Of-Speech (POS) Tagging, Chunking und Semantic Role Labeling (SRL). Der traditionelle Ansatz war hierbei, für jede dieser individuellen Aufgaben spezielle Features zu suchen (s.g. Feature-Engineering zu betreiben). Collobert et al. argumentieren, dass dieser Ansatz möglicherweise zu „engstirnig“ ist, und abstraktere oder kompliziertere Merkmale, die keiner klassischen linguistischen Theorie entsprechen, dadurch nie in Experimenten auftauchen. Zudem neigten klassische Systeme dazu, sich zu sehr auf die Merkmalsextraktion von Vorgängersystemen zu verlassen, was zu komplexen Laufzeitabhängigkeiten führte.

Deshalb entwickelten Collobert et al. Das s.g. SENNA-System. Dieses verfügt über zwei Arbeitsmodi, einen fensterbasierten Ansatz und einen satzbasierten Ansatz. Pascal beschränkte sich in seiner Präsentation auf letzteren. Hierbei wird jedes Wort eines Satzes der Länge N in eine Merkmalsmatrix der Form N*(50+x) eingesetzt; d.h. jedes Wort verfügt über 50 maschinenlernbare und x „fixe“, manuell eingefügte „Hilfsattribute“. Die 50 maschinenlernbaren Attribute entsprechen Word Embeddings. Die Fixen Attribute beinhalten z.B. Präfix, Suffix oder Pos-tag.

Collobert et al. beschränkten sich in der Generierung der Word Embeddings auf die häufigsten 100K Wörter aus dem Wall Street Journal Corpus. Hiermit erreichten sie über den gestellten Aufgaben (NER, POS, Chunking, SRL) F1 von 85%-97%, was jedoch zur Überholung der damaligen Rekorde noch nicht ganz ausreichte. Um die Leistung des Modells weiter zu steigern, wurde zusätzlich zum WSJ der Reuters-Korpus hinzugezogen.

Ein interessantes Detail ist hierbei, dass Collobert et al. aufrgund stark beschränkter Rechenressourcen zur Hyperparameteroptimierung auf ein progressives Feintuning bei wachsender Trainigskorpusgröße zurückgreifen mussten, eine “Brute force” Gittersuche war nicht praktikabel: Das System befand sich etwa 2 Monate im Training. Nach hinzunahme des Reuters Korpus war es Collobert et al. Nicht nur möglich, die Benchmarks in allen Aufgaben (außer SRL) zu schlagen, sondern auch erstmalig niedrigdimensionale Repräsentationen (Embeddings) für Terme zu generieren.

Student: Anton Serjogin

„Learning String Edit Distance“

Dieser Vortrag basiert auf dem Paper „Learning String Edit Distance” von Ristad, E. S., & Yianilos, P. N., 1999, und dem Probabilistischen Finite State Transducer, vorgestellt von Eisner 2002. Der Begriff “String Edit Distance” bedeutet Editierabstand. Er bezieht sich auf die Quantifikation der Unterschiedlichkeit zweier Strings, d.h. welche und wie viele Operationen (wahrscheinlich) notwendig wären, um einen String A in einen anderen String B zu konvertieren.

Dies hat Anwendungen in vielen Informationstheoretischen Gebieten, so zum Beispiel in der Computerlinguistik zur Rechtschreibkorrektur, oder in der Bioinformatik zur Klassifikation von Genen.

Im klassischen Sinne wird der Editierabstand mittels der s.g. “Levenshtein-Distanz” berechnet. Diese formuliert einen diskreten, deterministischen, determinierten, dynamischen Algorithmus zur Berechnung dieses Abstandes in der Domäne der ganzen Zahlen. Die Zahl repräsentiert hierbei die Anzahl an Lösch-, Einfüge-, Ersetzung- und Identitätsoperationen, die notwendig sind, um aus String A den String B zu formen. Folgende Beispiele illustrieren diesen Vorgang: Der Editierabstand zwischen “Carrot” und “Parrot” ist 1, da zur Umwandlung lediglich das C durch das P ersetzt werden muss. Der Editierabstand zwischen “Pirate” und “Climate” ist 3 (Ersetze “P” und “r”, Ergänze “C”).

Der Nachteil an dieser Vorgehensweise ist jedoch, dass der Levenshtein-Algorithmus keine Möglichkeit hat, Ambiguitäten zu “ranken”, oder bestimmte Transformationen probabilistisch zu bewerten. Um den Editierabstand probabilistisch zu formulieren, gibt es zwei Verfahren:

1. Der Viterbi-Editierabstand: Dieser misst die Wahrschienlichkeit einer Operation O_n+1 als Produkt der Wahrscheinlichkeiten der vorangegangenen Operationen O_1...O_n, d.h. p(O_1...O_n) = p(O_1) * p(O_2|O_1) … * p(O_n|O_1,…,O_n-1).
2. Der Stochastische Editierabstand: Dieser misst die Wahrscheinlichkeit p(O_1...O_n) als Anteil unter den Alternativen Editierfolgen (logprob).

Ein probabilistischer Transducer (Probabilistic Finite State Transducer, PFST), der zur Umsetzung des Viterbi-Editierabstandes benötigt wird, ist hierbei ein Endlicher Automat, der an jeder Kante ein Zeichen ausgeben kann. Außerdem ist jede Kante mit einer bestimmten Wahrscheinlichekit versehen. Zur Abschätzung dieser Wahrscheinlichkeiten setzte Eisner 2002 den Expectation Maximization (EM) Algorithmus ein.

Ein wichtiger Anwendungsfall für den Probabilistischen Editierabstand stellte Anton im Sinne der s.g. “Pronunciation Recognition” (“Ausspracheerkennung”/”PR”) vor. Hierbei gilt es, eine bestimmte Sequenz von phonologischen und phonetischen Phonemen in das entsprechende Wort zu übersetzen. Dies ist eine s.g. “Many-to-Many” Anwendung, d.h. ein Wort hat mehrere Aussprachen, und eine Aussprache kann verschiedene Wörter repräsentieren. Ein deterministischer Ansatz ist also schwer inherent missverständlich, Wahrschienlichkeitstheorie/Statistik liegt nah. Dies spiegelt sich in der Leistung der vorgestellten PR-Systeme wieder: Bei den phonetischen Übersetzungsaufgaben zeigten sowohl das Viterbi,- als auch das Statistische Editierabstandsmaß eine Fehlerrate von 9%, gegenüber einer Fehlerrate von etwa 50% bei der Annwendung der Levenshtein-Distanz.