Semantische Dekodierung von Sprachsignalen am Beispiel einer Mikrofonfeldsteuerung

Günther Wirsching und Matthias Wolff***

**Katholische Universität Eichstätt, **Brandenburgische Technische Universität*
guenther.wirsching@ku.de

1 Einleitung

In [5] wird gezeigt, dass – einige vernünftige Modellannahmen vorausgesetzt – jede natürliche Sprache endlich ist. Das gilt erst recht für domänenspezifische Sprachen wie beispielsweise verbale Maschinensteuerungen. Weiterhin folgen wir – wie in [5] – B. F. Skinners Auffassung, dass die Bedeutung einer sprachlichen Äußerung („verbales Verhalten“, *behavior*) nicht in dieser selbst, sondern im Eintreten bestimmter durch sie ausgelöster Auswirkungen (*consequences*) unter bestimmten Vorbedingungen (*antecedents*) liegt. Wir bezeichnen – geeignet formalisierte – Vorbedingungen und Auswirkungen als semantische Anker. Eine Bedeutung ist demnach formal durch eine Menge semantischer Anker beschrieben. Aus praktischen Gründen lassen wir zusätzlich auch Platzhalter (beispielsweise „dreistellige Kardinalzahl“ oder „Ortsbezeichnung“) zu und bezeichnen eine Bedeutung mit Platzhaltern als semantisches Schema [5]. Wie früher gezeigt wurde [4], eignen sich – in für die Modellierung der Mensch-Maschine-Schnittstelle interessanten Fällen – zu deren formaler Darstellung Merkmal-Werte-Relationen (*feature value relations*, FVR), das sind im einfachsten Fall beschriftete Bäume. Die Übersetzung zwischen Äußerung und als FVR dargestellter Bedeutung geschieht mit Hilfe sogenannter Äußerungs-Bedeutungs-Transduktoren (*utterance meaning transducers*, UMT). Die Äußerungsseite dieser Transduktoren kann aufgrund der oben stehenden Argumentation immer ein endlicher Akzeptor sein. Die genaue Natur der Bedeutungsseite ist Gegenstand unserer aktuellen Untersuchungen.

In diesem Beitrag stellen wir erste Versuche zur praktischen Konstruktion von UMTs für die Steuerung des Mikrofonfelds im Sprachlabor der BTU vor. Wir zeigen, dass obwohl die dazu erforderliche Sprache zweifellos endlich ist, man praktisch schon bei dieser einfachen Aufgabe äußerungsseitig nicht mit kreisfreien Grammatiken auskommt. Wir präsentieren weiterhin eine Zeichenkettendarstellung von Merkmal-Werte-Relationen und ein endliches Sprachmodell, welches diese FVR-Zeichenketten als Ausgabe erzeugt. Die FVR-Zeichenketten stellen einen ersten Zwischenschritt auf dem Weg zur automatischen Übersetzung von Erkennergebnissen in eine Baumstruktur dar. Die Umsetzung der syntagmatischen Relation zwischen den semantischen Ankern in eine hierarchische Relation gelingt allerdings nur teilweise, so dass für die Synthese des endgültigen Merkmal-Werte-Baums noch eine Nachbearbeitung notwendig ist. Wir zeigen außerdem, wie aus der Spracherkennerausgabe individuelle Konfidenzen für alle semantischen Anker ermittelt werden können.

Zukünftig wollen wir noch eine Übersetzung der FVR-Zeichenketten in Merkmal-Werte-Relationen realisieren. Vielversprechende Formalismen dafür scheinen Petrinetz-Transduktoren oder – falls wir uns auf Merkmal-Werte-Bäume beschränken können – Baumtransduktoren zu sein.

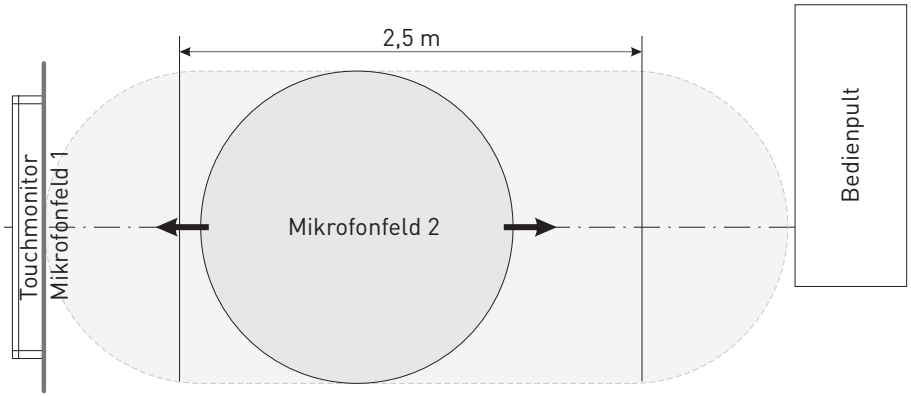

Abbildung 1 - Schematische Darstellung des zu steuernden Mikrofonfelds im Sprachlabor der BTU (Aufsicht, nicht maßstäblich).

2 Hintergrund

Das zu steuernde Mikrofonfeld im Sprachlabor der BTU besteht aus zwei Einheiten mit jeweils 32 Mikrofonen. Die erste Einheit ist um einen fest montierten 72-Zoll-Touchmonitor angeordnet, die zweite auf einem beweglichen Schlitten an der Decke des Labors in einem Winkel von ca. 81° zur ersten Einheit. Bild 1 zeigt die Anordnung schematisch (Details zum Mikrofonfeld siehe [1]). Die Sprachsteuerung dient zum Bewegen des Schlittens (Mikrofonfeld 2) sowie zum Ein- und Ausschalten einzelner Mikrofone oder von Mikrofongruppen.

In unserem Setting ist ein UMT dazu da,

1. zu einem gegebenen Kontext und einer gegebenen Bedeutungsstruktur möglichst genau alle passenden Wortfolgen abzubilden (als Akzeptor), und
2. zu jeder ankommenden passenden Wortfolge deren Bedeutung(en) auszugeben (als Generator).

Die bisherigen Arbeiten hierzu [3, 2] basieren auf der Idee, den ersten Teil als FST ohne Ausgaben und den zweiten Teil als PNT ohne Eingaben zu realisieren. Motiviert ist das von der Vorstellung, dass im ersten Teil ja nur *sequentielle* Daten zu verarbeiten sind, während der zweite Teil *partielle Worte* ausgeben muss.

2.1 Eine Hierarchie von Grammatiken

Unter einer *lokalen Grammatik* verstehen wir einen endlichen Akzeptor, der (möglichst) genau alle sinnvollen Sätze zu einem Themengebiet — häufig als *Domäne* bezeichnet — akzeptiert. (Oder, wenn die Kantenbeschriftungen als Ausgabezeichen interpretiert werden, generiert.)

Die *Bedeutung* einer Wortfolge wird als Baum B repräsentiert, wobei jedes Blatt eine Beschriftung aufweist, die einen konkreten semantischen Anker (keinen Platzhalter) bezeichnet. Stehen an einem oder mehreren Knoten eines beschrifteten Baums S Platzhalter, dann bezeichnen wir S als *semantisches Schema*. Wir sagen, dass eine Bedeutung B in ein semantisches Schema S *passt*, wenn B aus S durch die Ersetzung der Platzhalter durch semantische Anker entsteht.

Eine *mikrolokale* Grammatik zu einem gegebenen semantischen Schema S ist derjenige Teil einer lokalen Grammatik, der genau diejenigen Wortfolgen akzeptiert, deren Bedeutung ins semantische Schema S passt.

Eine *elementare* Grammatik zu einer gegebenen Bedeutung B ist derjenige Teil einer mikrolokalen Grammatik, der genau die Wortfolgen mit Bedeutung B akzeptiert.

Die in unserer Anwendung verwendeten semantischen Schemata sind:

`SWITCH`	Mikrofongruppen oder einzelne Mikrofone ein- oder aussschalten,
`MOVETOABS`	Bewegung des Schlittens an absolute Position und
`MOVETOREL`	Bewegung des Schlittens an Position relativ zu einem Bezugspunkt.

Alle enthalten sowohl Platzhalter für (untergeordnete) semantische Schemata, beispielsweise

`MCID`	Liste von Mikrofonnummern (in `SWITCH`),
`LDIM`	Längenangabe (in `MOVETOABS` und `MOVETOREL`)
usw.,	

als auch Platzhalter für konkrete semantische Anker.

2.2 Äußerung-Bedeutung-Transduktoren aus elementaren Grammatiken

Gemäß [3, 2] entsteht ein UMT als PNT durch Komponieren einer elementaren Grammatik zu einer gegebenen Bedeutung B mit einem Generator, der genau die Bedeutung B generiert — das Ergebnis nennen wir einen *elementaren UMT zur Bedeutung B*. Um zu einer mikrolokalen Grammatik einen UMT zu bauen, nimmt man einfach alle Bedeutungen der mikrolokalen Grammatik, baut zu jeder einzelnen Bedeutung einen elementaren UMT, und lässt anschließend eine geeignete PNT-Optimierung (noch zu bauen) darüber laufen.

Theoretisch wäre dagegen nichts einzuwenden — schließlich sind alle elementaren Grammatiken endlich, und in jedem Anwendungsfall sind auch nur endlich viele Bedeutungen zu verarbeiten. Matthias' Mikrofonfeldsteuerung enthält jedoch eine Teilgrammatik, bei der die Anzahl der möglichen Bedeutungen so groß ist, das diese Teilgrammatik praktisch nicht ohne Schleife realisierbar ist. Das Mikrofonfeld enthält 64 Mikrofone, von denen durch ein Kommando eine beliebige Teilmenge ein- oder ausgeschaltet werden können soll. Bekanntlich ist für jedes $n \in \mathbb{N}$ die Anzahl der Teilmengen der Menge $\{1,\ldots,n\}$ durch die folgende Formel gegeben:

$$t_n = \sum_{k=0}^{n} \binom{n}{k} = 2^n\,, \qquad \text{für } n = 64 \text{ also:} \quad 2^{64} = 2^4 \cdot 1024^6 \approx 1{,}84 \cdot 10^{19}\,.$$

Zudem sollte es prinzipiell möglich sein, in der Bedeutung auch die Reihenfolge der Elemente der Teilmenge beliebig festzulegen. Damit steigt die Anzahl der Möglichkeiten auf

$$r_n = \sum_{k=0}^{n} \binom{n}{k} \cdot k! = \sum_{k=0}^{n} \frac{n!}{(n-k)!} = n! \cdot \sum_{k=0}^{n} \frac{1}{k!} \approx e\,n!\,,$$

wobei $e = \sum_{k=0}^{\infty} \frac{1}{k!} \approx 2{,}718$ die Eulersche Zahl bezeichnet. Setzt man hier $n = 64$ ein, dann erhält man als Anzahl der Pfade in der Teilgrammatik

$$r_{64} \approx 3{,}4 \cdot 10^{89}\,,$$

was zunächst mal jenseits technischer Möglichkeiten liegt.

3 Der neue Plan

Wie zuvor sieht auch der neue Plan vor, einen UMT zu einer mikrolokalen Grammatik aus einem FST und einem PNT zu komponieren — diesmal allerdings nicht mit einem Zustand pro Bedeutung, sondern analog zur FST-Komposition mit einem Übergabestring, der alle relevante Information enthält.

3.1 Aufsammeln von Bedeutungen in der Grammatik

In der praktischen Implementierung ist der zweitgenannte Autor bereits den folgenden Weg gegangen: Die Bestandteile der Bedeutung werden bereits während der Erkennung aufgesammelt und in einem Format gespeichert, das ausreichend Information enthält und uns für die weitere Verarbeitung sinnvoll erscheint. Ein Teilsatz als Beispiel:

„ein Meter zwanzig“

Die Bedeutung dazu ist:

```
FVR[
  LDIM:0.61[
    val:1[
      DIG:1[
        o:1[
          (ein)1:1
  ] ] ] ][
    val:NaN[
      komma:NaN
  ] ][
    unit:0.7[
      (m)m:0.7
  ] ][
    val:0.36[
      CN2:0.36[
        t:0.36[
          (zwan)2(-zig):0.36
] ] ] ] ]
```

Diese Darstellung enthält in runden Klammern die erkannte Wortfolge (als „Kommentar“) sowie durch die eckigen Klammern und die Buchstabenfolgen dazwischen alle für Aufbau und Beschriftung des Baums erforderliche Information. Die Beschriftungen bestehen aus einem Namen und – abgeteilt durch einen Doppelpunkt – der Erkennerkonfidenz für das korrespondierende Signalstück. Man beachte, dass die Zeichenkette semantische Anker enthalten kann, zu denen kein Teil der Äußerung gehört (im Beispiel: `[val[komma]]`). Solche Ausgaben werden durch Kanten ohne Eingabesymbol in der mikrolokalen Grammatik erzeugt. Eine Konfidenz ist in diesem Fall nicht definiert (`NaN`).

Die formale Syntax gewichteter und kommentierter FVR-Zeichenketten ist (eBNF):

```
FVR      = "FVR" state
state    = "[" states statedef states "]"
states   = [ [ states ] state ]
statedef =  comments label comments [ ":" weight ] comments
comments = [ [ comments ] comment ]
comment  = "(" <alpha> ")"
label    = <alpha>
weight   = <numeric> | "NaN"
```

wobei `<alpha>` für beliebige Zeichenketten ohne Leer- und Sonderzeichen und `<numeric>` für Gleitkommazahlen (englisches Format) steht.

Bei der Verarbeitung werden zunächst die runden Klammern mit Inhalt herausgenommen und gesondert ausgegeben. Die verbleibende Zeichenkette hat die folgenden Eigenschaften:

a) Durch die Klammern ist die Baumstruktur eindeutig festgelegt.

b) Die Knotenbeschriftungen tauchen in der Reihenfolge auf, in der sie erzeugt werden. Prinzipiell ist bei gegebener Baumstruktur jedoch nicht jede Reihenfolge per Klammerausdruck darstellbar — dieser Umstand wird eine Extrabehandlung bei der Nachbearbeitung erfordern.

c) Die Klammern machen es prinzipiell möglich, einzelnen Bedeutungsbestandteilen Signalsegmente und damit Konfidenzen zuzuordnen. Hier besteht aber noch Forschungsbedarf, da wir noch keine Möglichkeit gefunden haben, eine Konfidenz allein mit Hilfe von im FST zulässigen Operationen zu berechnen.

3.2 Nachbearbeitung des Übergabestrings

Dem PNT-Teil des UMTs soll jetzt die Aufgabe übertragen werden, aus dem Übergabestring ein partielles Wort zu erzeugen. In Matthias' Aufbau werden dazu zwei Verarbeitungsschritte durchgeführt, von denen keiner per FST erledigbar ist:

1. Aufbau eines beschrifteten Baums aus der Klammerstruktur. Das hat Matthias versuchsweise in Java als rekursiv arbeitenden Parser implementiert; automatentheoretisch wäre dazu z. B. ein Kellerautomat geeignet. In der Java Implementierung ist es so, dass gleichnamige Geschwisterknoten in der gleichen Reihenfolge ausgegeben werden in der sie eingegeben wurden; damit wird implizit die Reihenfolge mitübergeben.

2. Kaskadieren gleichnamiger Geschwisterknoten unter der Verwendung ihrer Reihenfolge.

Hierbei ist zu beachten, dass es nicht genügt, vom ersten zum zweiten Schritt nur den Baum zu übergeben — die zur Kaskadierung benötigte Reihenfolge ist ebenfalls zu übermitteln. In Bild 2 sind die Verarbeitungsschritte an obigem Beispiel illustriert. In diesem Beispiel ist auch klar, dass es auf die Reihenfolge der „`val`"s ankommt.

Beim Experimentieren haben wir eine Möglichkeit gefunden, durch eine Erweiterung der Syntax des Übergabestrings das Verbaumen und das Kaskadieren in einem zu machen — leider funktioniert das nur, wenn sichergestellt ist, dass gleichnamige Beschriftungen im Übergabestring direkt hintereinander stehen. Wenn, wie im Beispiel, dazwischen eine andere Beschriftung „`unit`" steht, wäre das nur mit noch größeren Bauchschmerzen behandelbar.

4 Schluss

Aus unserer praktischen Arbeit an der Mikrofonfeldsteuerung lassen sich einige Schlüsse ziehen, die in der Entwicklung kognitiver Sprachdialogsysteme relevant sind:

1. Grundsätzlich sind FSTs auch geeignet, um Äußerungen in Bedeutungen zu übersetzen. Ein wichtiger Punkt hierbei ist die Möglichkeit, einzelnen Bestandteilen der Bedeutung konkrete Signalabschnitte zuzuordnen.

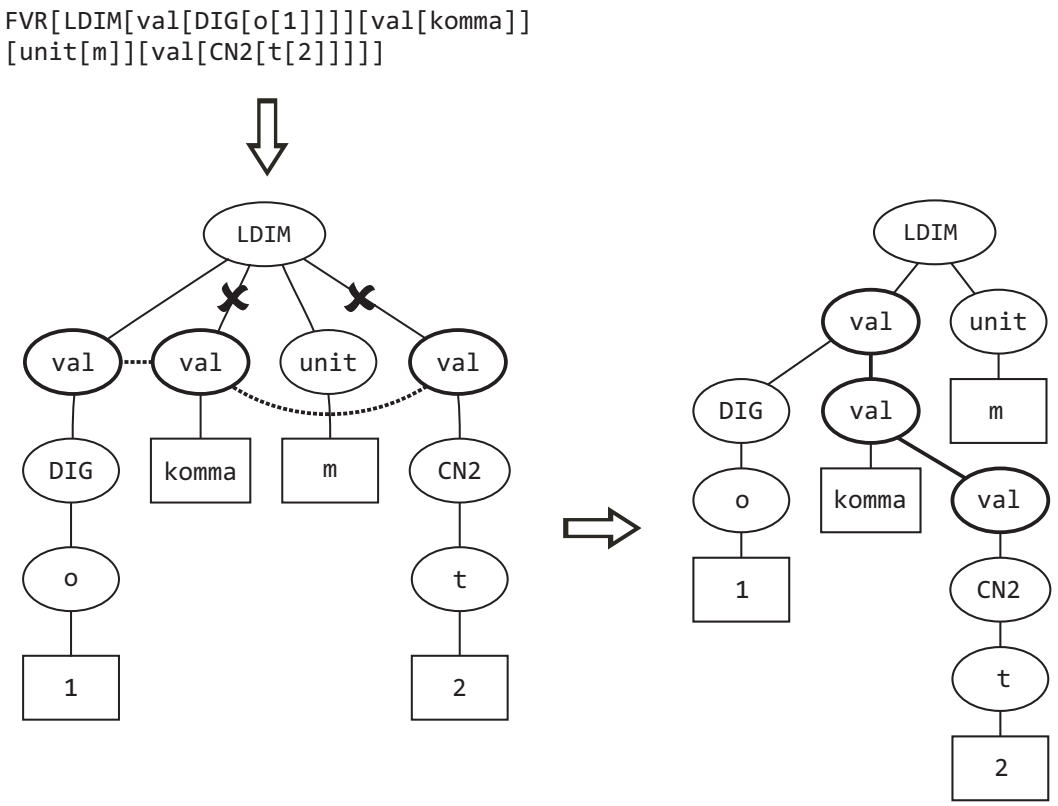

Abbildung 2 - Die beiden Verarbeitungsschritte in einem Beispiel.

2. Die große Zahl möglicher Bedeutungen in unserem Beispiel einer Mikrofonfeldsteuerung zeigt, dass eine Beschränkung auf azyklische Transduktoren nicht sinnvoll ist.

3. Mit unserer Konstruktion eines semantischen Übergabestrings scheint die Grenze dessen erreicht, was mit FSTs machbar ist. Weitere Verarbeitungsschritte müssen in der Lage sein, zumindest baumartige Strukturen angemessen zu verarbeiten.

Literatur

[1] M. Birth. Sprecherlokalisation mit einem 3d-mikrofonfeld. In *diesem Tagungsband*.

[2] R. Lorenz and M. Huber. Realising the Translation of Utterances into Meanings by Petri Net Transducers. In *ESSV2013*.

[3] Robert Lorenz and Markus Huber. Petri net transducers in semantic dialogue modelling. In Matthias Wolff, editor, *ESSV2012*, volume 64 of *Studientexte zur Sprachkommunikation*, pages 298–305, Dresden, August 2012. TUDPress.

[4] Günther Wirsching, Markus Huber, Christian Kölbl, Robert Lorenz, and Ronald Römer. Semantic dialogue modeling. In Anna Esposito, A. M. Esposito, A. Vinciarelli, R. Hoffmann, and V. C. Müller, editors, *Behavioral Cognitive Systems*, volume 7403 of *LNCS*, pages 104–113, 2012.

[5] Günther Wirsching and Robert Lorenz. Towards meaning-oriented language modeling. In *IEEE 4th Int. Conf. on Cognitive InfoCommunication, Budapest, Hungary, December 2-5*, pages 369–374, 2013.