

Parsing bidimensional de estructuras DAG mediante el algoritmo SCP y su aplicación al reconocimiento de habla

José F. Quesada
Universidad de Sevilla
jquesada@cica.es

Resumen: Este trabajo presenta una estrategia para la integración de los módulos de reconocimiento de habla, parsing basado en unificación y gestión de diálogo que permita la mejora de los sistemas de gestión de diálogo hablado. La estrategia propuesta se basa en la utilización del modelo basado en puntos de ruptura utilizado por el algoritmo SCP de análisis gramatical mediante propagación de restricciones. Este formato permite representar los retículos de palabras y su manipulación directa por parte del algoritmo SCP.

Palabras clave: Formalismos de análisis morfosintáctico, Sistemas de Diálogo Hablado, Aplicaciones del PLN

Abstract: This work describes an strategy for the integration of speech recognition, unification-based parsing and dialogue management and its application to spoken dialogue systems. The strategy is based on the use of the notion of breaking points implemented by the SCP (Syntactic Constraints Propagation) parsing algorithm. This format allows the representation of word lattices and their direct manipulation by SCP.

Keywords: Spoken Dialogue Systems, NLP Applications.

1 *Introducción: hacia la integración de sistemas de reconocimiento de habla, interpretación semántica y gestión de diálogo*

Durante los últimos años, en el campo de la Ingeniería del Lenguaje, se está asistiendo a la proliferación de múltiples aplicaciones que integran simultáneamente módulos o componentes de reconocimiento y síntesis de habla, análisis léxico-morfológico, gramatical e interpretación semántica, y gestión de diálogo. Entre estos entornos cabe citar Trindikit (Larsson et al 2000), Collagen (Rickel et al 2000), Communicator, Dhomme (Milward et al 2001, Quesada et al 2001), etc.

Independientemente de las arquitecturas o modelos de implementación elegidos para estas aplicaciones, es patente la utilización de distintos subsistemas especializados en cada una de las fases principales, básicamente:

- Reconocimiento/síntesis de habla;
- Interpretación semántica (lo que a su vez puede incluir análisis léxico-morfológico y gramatical); y

- Gestión de diálogo.

Uno de los principales retos a los que se enfrenta el diseño de este tipo de aplicaciones es la dificultad para la integración de los distintos componentes mencionados.

Básicamente, el sistema de reconocimiento de habla es un módulo aislado que usando como entrada la señal acústica genera la lista de los *n*-mejores candidatos reconocidos. Por su parte, el módulo de interpretación semántica, a partir del mejor candidato reconocido, construye la representación formal correspondiente. Ésta, a su vez, es transmitida al gestor de diálogo que se encarga de ejecutar las acciones pertinentes.

Este modelo, aunque fácilmente implementable, presenta graves deficiencias ya que cada uno de los módulos pierde la información que el resto de módulos podría suministrarle. En concreto, una integración más fuerte entre el reconocedor y el módulo de interpretación semántica permitiría dirigir el propio proceso de reconocimiento utilizando información lingüística de alto nivel (superior a la que habitualmente se utiliza para hacer el entrenamiento estocástico del reconocedor a

partir de un corpus –modelo de lenguaje–). Asimismo, la integración entre el módulo de interpretación semántica y el gestor de diálogo podría dirigir la interpretación usando las expectativas de diálogo generadas por el gestor de diálogo. Como consecuencia de la integración de los tres componentes, incluso el propio reconocimiento de habla podría estar dirigido o condicionado por las expectativas de gestión de diálogo.

En este ámbito existen algunos precedentes que han intentado aplicar técnicas de *parsing* para la mejora de la precisión de los sistemas de reconocimiento de habla:

- Aplicación de técnicas de *parsing* para seleccionar la secuencia más probable de un grafo de palabras que genera el reconocedor de habla (van Noord et al 1999).
- Chelba y Jelinek (1999) han utilizado un parser de izquierda a derecha para extraer los núcleos semánticos. Este sistema se ha usado para mejorar los modelos de tri-gramas usados para la creación de modelos de lenguaje.
- Khudanpur y Wu (1999) han utilizado un modelo similar al de Chelba y Jelinek que integra adicionalmente estructuras N-gram, estadísticas sobre palabras núcleo y dependencias de tema sobre un marco de máxima entropía. A este nivel resulta interesante la conclusión de estos autores:

“The use of syntactic structure in general and heads of syntactic constituents in particular has recently been shown to be beneficial for statistical language modeling. This paper provides an insightful analysis of this role of syntactic structure. It is shown that the predictive power of syntactic heads is mostly complementary to the predictive power of N-grams: they help in positions where an intervening phrase or clause separates the heads from the word being predicted, making the N-gram a poor predictor.” (Khudanpur y Wu 1999).

Este trabajo presenta una línea de investigación que pretende obtener una integración más fuerte entre los módulos de reconocimiento, interpretación semántica y gestión de diálogo. En concreto, el trabajo se centra específicamente en el primer paso de

integración: reconocimiento – interpretación semántica.

Para ello, la sección 2 presenta una formalización de los procesos y componentes que intervienen, así como un análisis de los principales problemas y estrategias de solución, cuya conclusión principal es la necesidad de obtener un nuevo algoritmo de *parsing* para gramáticas libres (o independientes) de contexto que permita la manipulación directa de estructuras bidimensionales (específicamente grafos acíclicos dirigidos que permitan la representación de los *word lattices* generados por los reconocedores de habla) como entrada de análisis gramatical, así como las expectativas de diálogo como estructuras que filtren o dirijan el proceso de *parsing*.

El algoritmo SCP (Syntactic Constraint Propagation) de análisis para gramáticas libres de contexto permite abordar las dos estrategias de integración anteriores. La sección 3 presenta las principales características de este algoritmo.

Por último, la sección 4 describe las estrategias básicas de adaptación del algoritmo SCP para lograr la integración con sistemas de reconocimiento de habla.

2 Formalización del problema de la integración

2.1 Reconocimiento de habla

Un reconocedor de habla (SRer: *Speech Recogniser*) se puede describir genéricamente como una función –f– definida a partir de los patrones acústicos del léxico de referencia (APat: *Acoustic Patterns*) y un modelo de lenguaje (LMod: *Language Model*):

$$SRer = f(APat, LMod)$$

Por su parte, un sistema de reconocimiento de habla (SRSys: *Speech Recognition System*), concebido como un proceso, es una función –g– que usando el reconocedor (SRer) y la entrada acústica (AInput: *Acoustic Input*) recibida en un momento determinado genera internamente una representación del proceso en forma de retículo de palabras (WL: *Word Lattice*):

$$\begin{aligned} SRSys &= g(SRer, AInput) \\ &= g(f(APat, LMod), AInput) \\ &=> WL \end{aligned}$$

2.2 Interpretación semántica

De cara a los objetivos de este trabajo, el módulo de interpretación semántica se describe como un sistema de *parsing* para gramáticas

libres de contexto ampliado con unificación. Es decir, se sigue el paradigma de las gramáticas de unificación.

En este marco, un parser para gramáticas libres de contexto basado en unificación (UPer: *Unification-based CFG Parser*) se define como una función –l– generada a partir del conjunto de reglas libres de contexto (CFGGrammar: *Context Free Grammar*) y las reglas de unificación (URules: *Unification rules*):

$$UPer = l(CFGGrammar, URules)$$

Un sistema de *parsing* basado en unificación (UPngSys: *Unification-based CFG Parsing System*) se define como una función –m– que utilizará el parser (UPer) y una oración de entrada (S: *Sentence*) para generar como salida una representación semántica de la oración (SISent: *Semantically Interpreted Sentence*) según el modelo de representación formal elegido.

$$\begin{aligned} UPSys &= m(UPer, S) \\ &= m(l(CFGGrammar, URules), S) \\ &=> SISent \end{aligned}$$

2.3 Gestión de diálogo

Siguiendo con la misma estrategia anterior, definiremos un gestor de diálogo en tanto que componente (DMer: *Dialogue Manager*) como una función –r– generada a partir de una especificación de modelo de diálogo (DSpec: *Dialogue Specification*):

$$DMer = r(DSpec)$$

Por otro lado, consideraremos un sistema de gestión de diálogo (DMSys: *Dialogue Management System*) como una función –s– que utilizará como entrada tanto el gestor de diálogo como la oración interpretada (SISent) obtenida por el módulo de interpretación semántica y la historia del diálogo (DHis: *Dialogue History*) para generar un conjunto de acciones de diálogo (DAct: *Dialogue Actions*) y de expectativas (Expt: *Expectations*):

$$\begin{aligned} DMSys &= s(DMer, SISent, DHis) \\ &= s(r(DSpec), SISent, DHis) \\ &=> DAct \ \& \ Expt \end{aligned}$$

2.4 Problema 1: WL - S

Utilizando la formalización de componentes y procesos anteriores, podemos observar cómo aparece un primer problema en la interfaz entre el sistema de reconocimiento de habla, cuya salida natural es un *word lattice* (WL) y la entrada al sistema de *parsing*, que espera recibir

una oración o cadena unidimensional de palabras.

Este problema se resuelve habitualmente recurriendo a algoritmos de tipo *n-best* que obtienen las secuencias de palabras del *word lattice* con mayor probabilidad acústica acumulada. No obstante, este proceso pierde en gran parte la motivación lingüística y genera resultados gramatical y semánticamente incorrectos.

La alternativa consistente en utilizar como entrada al *parser* todas las posibles secuencias contenidas en el *word lattice* es prácticamente inviable teniendo en cuenta que el número de combinaciones contenidas en éste es varios órdenes de magnitud superior al número de palabras.

La estrategia que proponemos se basa en la modificación del algoritmo de *parsing* de forma que permita la entrada directa de la información contenida en el *word lattice*. Aunque las ventajas son obvias al conseguir un enlace fuerte entre los módulos de reconocimiento y de interpretación semántica, los retos que esta solución debe afrontar son complejos ya que es necesario obtener un nuevo modelo formal de *parsing* que además sea lo suficientemente eficiente como para manipular la cantidad de información del *word lattice*.

2.5 Problema 2: UPng - Expt

El segundo problema de integración aparece entre los módulos de interpretación semántica y de gestión de diálogo. El problema se puede caracterizar por la falta de conocimiento por parte del módulo de interpretación semántica acerca de la situación de diálogo.

Para solventar esta desconexión se propone como estrategia la utilización de las expectativas de diálogo como información adicional durante el proceso de interpretación semántica (*parsing* y unificación).

Esta información permitirá al módulo de interpretación semántica desambiguar construcciones ambiguas utilizando las expectativas para dirigir la selección de la estrategia de análisis. Por otra parte, este mecanismo no es destructivo como podría ser la utilización de estrategias de análisis por sub-gramáticas que realmente rechaza los análisis no contemplados en la sub-gramática.

Al igual que para el caso anterior, los principales retos que plantea esta estrategia se centran en la formalización de un nuevo modelo

de *parsing* que incorpore las expectativas de diálogo como información de entrada adicional durante el proceso de *parsing*, así como la obtención de una implementación suficientemente eficiente para permitir su aplicación en sistemas reales de gestión de diálogo hablado.

2.6 Un nuevo modelo de *parsing* para word-lattices con integración de expectativas de diálogo

Como resumen de esta sección, se puede considerar que la solución propuesta pasa por la utilización de una nueva función m para el proceso de *parsing* basado en unificación (UPSys):

$$\begin{aligned} \text{UPSys} &= m'(U\text{Per}, \text{WL}, \text{Expt}) \\ &= m'(l(\text{CFG}\text{Grammar}, \text{URules}), \\ &\quad \text{WL}, \text{Expt}) \\ &\Rightarrow \text{SIsent} \end{aligned}$$

3 El algoritmo SCP de análisis gramatical mediante propagación de restricciones sintácticas

SCP (Quesada 1997a) es un algoritmo de *parsing* para gramáticas libres de contexto que logra una gran eficiencia computacional manteniendo la motivación lingüística.

Las principales características del algoritmo son:

- Implementa una estrategia de *parsing* bidireccional, ascendente y dirigido por eventos.
- Incorpora un fuerte mecanismo de predicción descendente que permite un filtrado de eventos en fases tempranas del *parsing*, basado en las relaciones de derivabilidad parcial y adyacencia.
- Está basado en una capa computacional diseñada específicamente para la manipulación de objetos de tipo simbólico.

De forma paralela, para cada apartado anterior se pueden destacar las siguientes consecuencias a los niveles formal, lingüístico y computacional del algoritmo SCP:

- Formalmente, el algoritmo evita los fenómenos de *overparsing* (Quesada 1998), y se ha obtenido una demostración de los teoremas de corrección y completitud para gramáticas independientes de contexto sin restricciones.

- Lingüísticamente, el algoritmo manipula de una forma robusta y eficiente todos los fenómenos independientes de contexto, incluyendo recursión por la izquierda, gramáticas cíclicas, dependencias no locales, etc.
- Computacionalmente, el algoritmo ha conseguido un alto nivel de eficiencia. Aunque la complejidad algorítmica es n^3 en el peor caso, para fenómenos habituales de las lenguas naturales como recursión, dependencias locales, no locales, etc., el algoritmo SCP presenta una complejidad lineal (n).

4 Parsing bidimensional de estructuras DAG mediante SCP

4.1 Puntos de ruptura (Breaking points)

Una de las ideas más interesantes del algoritmo SCP es la utilización en el interfaz de entrada de un modelo de representación basado en estructuras denominadas puntos de ruptura (*breaking points*).

Habitualmente, durante el diseño de algoritmos de *parsing* se considera que la entrada es una cadena de caracteres donde las propias palabras (separadas por espacios en blanco u otros separadores) se corresponden unívocamente con *tokens*. No obstante, la consideración de fenómenos léxico-morfológicos como clíticos (“*díselo*”) estructuras multipalabra (“*sin embargo*”), nombres compuestos (“*República de Argentina*”), etc., necesita incorporar un mecanismo más sofisticado para la *tokenización* del nivel léxico-morfológico.

Para realizar esta función, el algoritmo SCP utiliza en la entrada un formato de representación basado en puntos de ruptura (*breaking points*). El analizador léxico-morfológico debe segmentar la cadena de entrada marcándola en los puntos de ruptura que considere necesarios (el caso base o ingenuo haría corresponder cada punto de ruptura con una separación entre palabras).

Por ejemplo, si consideramos la siguiente frase:

Sin embargo, díselo.

Podríamos generar la siguiente secuencia de puntos de ruptura:

0 Sin 1 embargo 2, 3 dí- 4-se- 5-lo 6. 7

Donde los guiones se han incorporado para indicar que hay una conexión entre las secuencias de caracteres situadas a cada lado del punto de ruptura.

Utilizando este mecanismo, la entrada al algoritmo SCP se representa como una lista de estructuras del tipo:

<fbp,lbf,word,CAT,FDs>

Donde:

- Fbp (*first breaking point*): indica el punto de ruptura que marca el principio del token.
- Lbp (*last breaking point*): indica el punto de ruptura que marca el final del token. Evidentemente, fbp tiene que ser estrictamente menor que lbp, pero no necesariamente tienen que ser posiciones consecutivas.
- Word: es la concatenación de todas las secuencias de caracteres situadas entre fbp y lbp.
- CAT (*category*): indica la categoría sintáctica o símbolo preterminal obtenido por el analizador léxico para la secuencia word. Si una misma secuencia posee más de una categoría (ambigüedad léxica) aparecerá una estructura diferente para cada categoría.
- FDs (*functional descriptions*): es la lista de todas las descripciones funcionales (estructuras de rasgos) correspondientes a la palabra word con la categoría CAT. En caso de ambigüedad estructural una misma palabra word usando la categoría CAT podrá ir asociada a múltiples estructuras FD. En este trabajo nos centraremos fundamentalmente en la capacidad de análisis gramatical de estructuras bidimensionales tipo *word lattice* y no prestaremos especial atención a la parte de unificación de este proceso. Por tanto, en los ejemplos siguientes prescindiremos de la parte FDs de la representación.

Teniendo en cuenta este modelo, la representación para la frase anterior podría ser:

```
<0,1,sin,prep>
<1,2,embargo,n>
<1,2,embargo,v>
<0,2,sin embargo,conj>
<2,3,"",",coma>
<3,4,dí-,v>
...
```

La estrategia bidireccional de SCP junto con el mecanismo de análisis basado en nodos y eventos conectados mediante estructuras CaD (Quesada 1997a, Quesada 1997b) permite una

manipulación eficiente de esta entrada. De hecho, SCP crea una estructura CaD para cada punto de ruptura. Las estructuras CaD incluyen campos diferentes para representar los nodos enlazados por la izquierda y derecha del punto de ruptura, no existiendo por tanto una conexión fuerte entre la secuencia de palabras y los puntos de ruptura. Así pues, cada nodo puede ser asociado a estructuras CaD no consecutivas, lo que permite la utilización del modelo de representación basado en puntos de ruptura.

4.2 Retículos de palabras y puntos de ruptura

Aunque existen varios modelos para la representación del retículo de palabras (*word lattice*) obtenido por el reconocedor de habla, podemos considerar que a grandes líneas dicho retículo es una lista de estructuras, cada una de las cuales se representa como:

<PI, PF, W, Ias>

donde:

- PI (punto inicial): indica el punto de inicio de la palabra reconocida en la señal acústica.
- PF (punto final): indica el punto de finalización de la palabra reconocida.
- W (word): es la palabra reconocida en el intervalo definido por PI y PF.
- Ias (información acústica): incluye uno o más valores acerca de las probabilidades acústicas asociadas con la palabra reconocida (probabilidad aislada, acumulada, etc.).

Se puede observar que existe un claro paralelismo entre la estructura de la información contenida en un *word lattice* y el modelo de representación basado en breaking points utilizado por SCP.

De esta forma resulta trivial establecer un proceso de transferencia de la información contenida en el formato word-lattice hasta su representación en breaking points utilizando como proceso intermedio un analizador léxico-morfológico que incorpore durante el proceso la información sobre categoría y descripciones funcionales.

Asimismo, es necesario incrementar el modelo de puntos de ruptura para incorporar la información acústica, la cual puede ser útil para incorporar mecanismos de desambiguación que utilicen la probabilidad acústica del propio reconocedor. De esta forma, la integración no

sólo incorpora información lingüística al reconocimiento de habla (en realidad *parsing* basado en unificación del *word lattice*) sino que el propio análisis sintáctico y semántico incorpora información acústica del reconocedor.

4.3 Consideraciones de eficiencia

El modelo propuesto permite realmente el análisis exhaustivo de todas las alternativas contenidas en el *word lattice*. De esta forma, la estrategia propuesta debe compararse con el enfoque basado en el análisis de todas las secuencias de palabras contenidas en el *word lattice*.

Para este último modelo se puede observar que la complejidad computacional es exponencial. Por ejemplo, si suponemos un simple retículo de palabras donde en cada punto del intervalo de reconocimiento empiecen y terminen 2 palabras, el número total de combinaciones de palabras sería 2^n (complejidad exponencial), donde n sería el número de puntos del intervalo de reconocimiento.

Por otra parte, la complejidad computacional para la estrategia basada en la integración con SCP realmente utiliza una única cadena de entrada, y, por tanto, se corresponderá con la complejidad del propio algoritmo SCP, que es n^3 para el peor de los casos.

En el ejemplo considerado, el número de estructuras CaD sería $2 * n$ (donde n es el número de puntos del intervalo de reconocimiento), manteniéndose la complejidad como n^3 (polinómica).

Para evaluar la eficiencia real, se ha implementado un prototipo que implementa la estrategia explicada anteriormente. El prototipo, implementado en C y ejecutándose sobre una máquina con procesador Pentium III 500 Mhz, 32MB de memoria y SO Linux RedHad 7.1, usando *word lattices* para frases reales de una longitud de entre 5 y 20 palabras (tamaño del *word lattice* de entre 500 y 2000 items reconocidos), ha obtenido tiempos de *parsing* (incluida la integración) comprendidos entre 0.01 y 0.90 segundos, es decir, entre centésimas y décimas de segundo.

5 Conclusión y trabajo futuro

En este trabajo se ha propuesto una estrategia para la integración fuerte entre los módulos de reconocimiento de habla y de interpretación

semántica (*parsing* basado en unificación) en los sistemas de gestión de diálogo hablado. La estrategia se basa en la utilización del algoritmo SCP, y en concreto en el mecanismo de representación basado en los puntos de ruptura, el cual permite el tratamiento de la información contenida en los retículos de palabras generados por los reconocedores de habla.

Aunque los niveles de complejidad computacional como eficiencia real han sido excelentes en el prototipo implementado siguiendo la estrategia propuesta, como línea de trabajo futuro es necesario abordar mecanismos sofisticados de compactación del propio *word-lattice* que aceleren el proceso de *parsing*.

Bibliografía

- Boros, M., Eckert W., Gallwitz, F., Görz, G., Hanrieder, G., Niemann, H. 1996. Toward Understanding Spontaneous Speech: Word Accuracy vs. Semantic Accuracy. *Proceedings of ICSLP-96 - International Conference on Spoken Language Processing*, Philadelphia, Oct. 3-6, p. 1005-1008
- Chelba, C., Jelinek, F. 1999. Recognition performance of a structured language model. *Proceedings of Eurospeech*, 1999, pp. 1567-1570.
- Chow, Y. L., Roukos, S. Speech Understanding using a unification grammar. *Proceedings of the IEEE International Conference on Acoustic, Speech and Signal Processing*. Glasgow, Scotland, pp. 727-730.
- Khudanpur, S., Wu, J. 1999. A Maximum Entropy Language Model to Integrate N-Grams and Topic Dependencies for Conversational Speech Recognition. *Proceedings of ICASSP'99*, pp. 553-556.
- Larsson, S., Berman, A., Bos, J., Grönqvist, L., Lunglöf, P., Traum, D.: 2000. *TRINDIKIT 2.0 Manual -revised version*. Trindi project, Deliverable 5.3. <http://www.ling.gu.se/projekt/trindi/publications.html>
- Mangu, L. et al. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language Magazine*. **14(4)**, 373-400.

- Milward, D., Thomas, J., Amores, J.G., Bos, J., Boye, J., Cooper, R., Ericsson, S., Gorrell, G., Hjelm, D., Holt, A., Knight, S., Larsson, S., Lewin, I., Quesada, J.F. 2001. *A D Homme Demonstrator in English, Swedish and Spanish*. Dhomme project. Deliverable 2.2. <http://www.ling.gu.se/projekt/dhomme/Publications/publications.html>
- Moore, R., Appelt, D., Dowding, J., Gawron, J., Moran, D. 1995. Combining Linguistic and Statistical Knowledge Sources in Natural Language Processing for ATIS. *Proceedings of the ARPA Spoken Language Systems Technology Workshop*, Austin.
- Quesada, J. F. 1997a. *El algoritmo SCP de análisis sintáctico mediante propagación de restricciones*. Tesis Doctoral. Universidad de Sevilla.
- Quesada, J. F. 1997b. A General, Sound and Efficient Natural Language Parsing Algorithm based on Syntactic Constraints Propagation. *Proceedings of the VII Conference AEPIA 97*, pp. 775-786.
- Quesada, J. F. 1998. Bidirectional and Event-Driven Parsing with Multi-Virtual Trees. En Martín-Vide, ed. *Mathematical and Computational Analysis of Natural Language*, John Benjamins, pp. 253-265.
- Quesada, J. F. 1999. Overparsing. En Martín-Vide, ed. *Issues in Mathematical Linguistics*, John Benjamins, pp. 165-182.
- Quesada, J. F., Amores, J. G. 2000. *Diseño e Implementación de Sistemas de Traducción Automática*. Universidad de Sevilla.
- Quesada, J. F., Amores, J. G., Bos, J., Ericsson, S., Gorrell, G., Knight, S., Lewin, I., Milward, D., Rayner, M. 2001. *Configuring Linguistic Components in a Plug and Play Environment*. Dhomme project. Deliverable 3.1. <http://www.ling.gu.se/projekt/dhomme/Publications/publications.html>
- Rickel, J., Ganeshan, R., Rich, C., Sidner, C.L., Lesh, N. 2000. Task-Oriented Tutorial Dialogue: Issues and Agents. *AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications*. Technical Report FS-00-01, November 2000, pp 52-57.
- Schmid, L. 1994. Parsing word graphs using a linguistic grammar and a statistical language model. *Proceedings of ICASSP-94*, vol. II, pp. 41-44.
- Ward, W. 1990. *Understanding spontaneous speech*. Carnegie-Mellon University Computer Science Technical Report.
- Van Noord, G., Bouma, G. Koeling, R. Nederhof, M.J. 1999. Robust Grammatical Analysis for Spoken Dialogue Systems. *Journal of Natural Language Engineering*, **5(1)**, pp. 45-93.
- Weber, H., Spilker, J., Görz, G: Parsing N Best Trees from a Word Lattice. *Advances in Artificial Intelligence. Proceedings of KI-97*, Freiburg, 9.-12.9.1997. Berlin: Springer (LNAI 1303), 1997, p. 279-288