Phase 2 of the D4 Project

Helmut Schmid and Sabine Schulte im Walde
Statistical Verb-Clustering Model

- soft clustering: Verbs may belong to several clusters trained on verb-argument tuples
- clusters together verbs with similar subcategorization and selectional restriction properties
- assumption: The arguments are independent of the verb given the cluster.
- probability of the tuple \((\text{read}, \text{SUBJ:OBJ}, \text{Peter}, \text{book})\)

\[
\sum_c p(c)p(\text{read}|c)p(\text{SUBJ:OBJ}|c) \ p(\text{Peter}|c, S:O, 1) \ p(\text{book}|c, S:O, 2)
\]

- Selectional restrictions are expressed with WordNet concepts r.

\[
p(\text{book}|\text{slot}) = \sum_r p(r|\text{slot})p(\text{book}|r)
\]

- training with Expectation-Maximization algorithm and the Minimum-Description-Length principle
Main Goals

- **Verb Clustering**
  - Extension of the verb-clustering model to deal with *adjuncts*
  - Induction of *noun taxonomies*

- **Parsing**
  - Usage of *very large amounts of context information* for syntactic disambiguation
  - Selection of the *most relevant* subset of context information with *decision trees*
The current model only deals with arguments.

Verb-argument tuples extracted from automatically parsed data also contain adjuncts (which are currently treated like arguments).

⇒ A more adequate modelling of adjuncts is likely to improve the verb-clustering model.
⇒ Automatic distinction between arguments and adjuncts
⇒ New insights into into the difference between arguments and adjuncts
Observations (Schütze, Merlo)

- Arguments are verb-specific; adjuncts combine with a wide range of verbs.
  
  read/sleep/... in the garden – believe in god

- Only the first one of a sequence of PPs is a possible argument. The others must be adjuncts.
  
  John waited for Sandy in the hotel at the bar until 9 o’clock.
Extension of the Model

Generation of the tuple \((C5, \text{read}, \text{SUBJ:NP}, \text{Peter}, \text{book}, \text{yesterday})\)

1. Choose a **verb cluster** \(\Rightarrow p(C5)\)
2. Choose a **verb** given cluster C5 \(\Rightarrow p(\text{read}|C5)\)
3. Choose a **subcat frame** given cluster C5 \(\Rightarrow p(\text{SUBJ:NP}|C5)\)
4. Choose a **filler** for each slot given the cluster, the frame, and the slot number
   \(\Rightarrow p(\text{Peter}|C5, \text{SUBJ:NP}, 1)\)
   \(\Rightarrow p(\text{book}|C5, \text{SUBJ:NP}, 2)\)
5. Decide whether to add an **adjunct** (yes)
   \(\Rightarrow p(\text{adjunct})\)
6. Choose an **adjunct** filler
   \(\Rightarrow p(\text{yesterday})\)
7. Decide whether to add another **adjunct** (no)
   \(\Rightarrow 1\)

Assumption: Adjuncts are independent of the verb (cluster). They freely combine with any verb.
Extension of the Model

Generation of the tuple \((C5, \text{read}, \text{SUBJ:NP}, \text{Peter, book, yesterday})\)

1. Choose a verb cluster \(\Rightarrow p(C5)\)
2. Choose a verb given cluster \(C5\) \(\Rightarrow p(\text{read}|C5)\)
3. Choose a subcat frame given cluster \(C5\) \(\Rightarrow p(\text{SUBJ:NP}|C5)\)
4. Choose a filler for each slot given the cluster, the frame, and the slot number \(\Rightarrow p(\text{Peter}|C5, \text{SUBJ:NP}, 1)\), \(\Rightarrow p(\text{book}|C5, \text{SUBJ:NP}, 2)\)
5. Decide whether to add an adjunct (yes) \(\Rightarrow p_{\text{adjunct}}\)
6. Choose an adjunct filler \(\Rightarrow p(\text{yesterday})\)
7. Decide whether to add another adjunct (no) \(\Rightarrow 1 − p_{\text{adjunct}}\)

Assumption: Adjuncts are independent of the verb (cluster). They freely combine with any verb.
From a parse of the sentence

John waited for Mary for half an hour.

without an argument/adjunct distinction, the following underspecified tuple is extracted:

(wait, \{SUBJ:PP, SUBJ\}, John, for Mary, for hour)

The verb-clustering model is trained on a corpus of underspecified (as well as on fully specified) tuples.

**Expectation:** The trained model will assign a higher probability to the tuple with the frame \text{SUBJ:PP}, and thereby will correctly resolve the argument/adjunct ambiguity.
Induction of Noun Hierarchies
Semantic realization of a predicate’s complement

Reference to the syntactic function and the thematic role

Example: *drink tea, drink coffee, drink beer*, etc.

\[ \rightarrow \text{drink a beverage} (\rightarrow \text{drink a substance}) \]

Preference: degree of acceptability

Requires inventory (and organisation) of semantic categories

\[ \rightarrow \text{clusters / WordNet} \]
Available sources of selectional preferences:
- WordNet (used by most work)
- Cluster (few)
- Similarity-based approach (only one)

Question:
Can we induce selectional preferences using a noun hierarchy that is NOT manually pre-defined?

Goals:
1. Automatic induction of noun hierarchy
2. Application to selectional preferences (in D4 verb clustering model)
Approaches: Overview

- Cluster-based selectional preferences
  - Pereira et al. (1993)
  - Rooth et al. (1999), Schulte im Walde et al. (2008)

- WordNet-based selectional preferences
  - Resnik (1997) - association strength
  - Li/Abe (1998) - MDL cut
  - Abney/Light (1999) - HMM
  - Ciaramita/Johnson (2000) - Bayesian belief network
  - Clark/Weir (2002) - MDL cut
  - Light/Greiff (2002) - summary of approaches
  - Brockmann/Lapata (2003) - comparison
WordNet-based Preferences

<entity>

<object, inanimate object, physical object>

<substance, matter>

<fuel> <food, nutrient> <body substance>

<beverage, drink, potable>

<tea> <coffee, java> <milk>

direct objects of drink
Katrin Erk (2007): “A simple, similarity-based model for selectional preferences”. In: *Proceedings of the 45th Conference of the Association for Computational Linguistics*

- Alternative approach to WordNet-based models
- Basis: corpus-based semantic similarity metrics $\rightarrow$ generalise from seen headwords to other, similar words
- Independence of manual lexical resource
- Corpus for computing the similarity metrics can be freely chosen, allowing domain variation
- Application: semantic roles
Primary corpus: extract tuples \(<p,r,w>\) of a predicate \(p\), an argument position \(r\), and a seen headword \(w\).

Generalisation corpus: compute a corpus-based semantic similarity metric.

Selectional preference \(S\) of a functional relation \(r\) for a possible headword \(w_0\) is modelled as a weighted sum of the similarities between \(w_0\) and the seen headwords \(w\):

\[
S_r(w_0) = \sum_{w \in \text{Seen}(r)} \text{sim}(w_0, w) \alpha(w)
\]

Metrics: mutual information, cosine, dice, etc.

Weights \(\alpha\): uniform, inverse document frequency, etc.
Approach by Erk (2007)

- Selectional preferences computed for frame-semantic roles, e.g., what is the selectional preference for the **Vehicle** role of **Ride Vehicle**?
- Primary corpus: FrameNet data
- Generalisation corpus: British National Corpus

**Ride_vehicle:** **Vehicle**
**Jaccard:** truck, boat, coach, van, ship, lorry, creature, flight, guy, carriage, helicopter, lad
**Cosine:** it, there, they, that, i, ship, second one, machine, e, other one, response, second

**Ingest_substance:** **Substance**
**Jaccard:** loaf, ice cream, you, some, that, er, photo, kind, he, type, thing, milk
**Cosine:** there, that, object, argument, them, version, machine, result, response, item, concept, s
Parts:

1. Create classes of synonymous/similar nouns
   - rely on lexico-syntactic patterns (Hearst, 1992)
   - rely on word graphs (Dorow, 2007)
   - rely on similarity measures (similar to Erk, 2007)
   - use standard clustering techniques

2. Organise classes in is-a hierarchy
   - rely on lexico-syntactic patterns (Hearst, 1992)
   - use word graphs (Dorow, 2007) for disambiguation
   - specify similarity measures for hypernym relation
Syntactic Disambiguation with Very Large Contexts in a Statistical Parser
A statistical parser computes the most probable parse tree.

The parse tree probability is the product of the rule probabilities.

The probability of a grammar rule such as $VP \rightarrow VP, VP, CC VP$ is often decomposed into a product of symbol probabilities:

$$p(VP \mid VP \rightarrow ...) \times p(, \mid VP \rightarrow VP ...) \times p(VP \mid VP \rightarrow VP, ...) \times p(, \mid VP \rightarrow VP, VP ...) \times p(CC \mid VP \rightarrow VP, VP, ...) \times p(VP \mid VP \rightarrow VP, VP, CC ...)$$

**Crucial task:** accurate estimation of these probabilities
POS Tagging

Same task, different context:

A part-of-speech tagger finds the most probable POS sequence for a given word sequence:

Das ART.Def.Nom.Sg.Neut
zu PART.Zu
versteuernde ADJA.Pos.Nom.Sg.Neut
Einkommen N.Reg.Nom.Sg.Neut
sinkt VFIN.Full.3.Sg.Pres.Ind

HMM POS taggers compute the probability of a POS tag sequence which is decomposed as follows:

\[ p(\text{ART.Def.Nom.Sg.Neut}) \times p(\text{PART.Zu} \mid \text{ART.Def.Nom.Sg.Neut}) \times p(\text{ADJA.Pos.Nom.Sg.Neut} \mid \text{ART.Def.Nom.Sg.Neut, PART.Zu}) \times \ldots \]
Given a fine-grained POS tagset and a large context size, many grammatical POS sequences are rare or not observed at all.

The POS trigram ART.Def.Nom.Sg.Neut, PART.Zu, ADJA.Pos.Nom.Sg.Neut e.g. does not occur in the Tiger treebank.

⇒ The trigram probability is 0.
⇒ The best tag sequence for the example sentence Das zu versteuernde Einkommen sinkt cannot be found.
Solution proposed in Schmid/Laws 2008:

- Replace each POS tag by an attribute vector
  \[ \text{ADJA.Pos.Nom.Sg.Neut} \rightarrow (\text{ADJA}, \text{Pos}, \text{Nom}, \text{Sg}, \text{Neut}) \]

- Decompose the tag probability into a product of attribute probabilities

\[
p(\text{ADJA.Pos.Nom.Sg.Neut} \mid \text{ART.Def.Nom.Sg.Neut}, \text{PART.Zu}) = \]
\[
p(\text{ADJA} \mid \text{ART_Def.Nom.Sg.Neut}, \text{PART.Zu})
\]
\[
p(\text{Pos} \mid \text{ART_Def.Nom.Sg.Neut}, \text{PART.Zu}, \text{ADJA})
\]
\[
p(\text{Nom} \mid \text{ART_Def.Nom.Sg.Neut}, \text{PART.Zu}, \text{ADJA.Pos})
\]
\[
p(\text{Sg} \mid \text{ART_Def.Nom.Sg.Neut}, \text{PART.Zu}, \text{ADJA.Pos.Nom})
\]
\[
p(\text{Neut} \mid \text{ART_Def.Nom.Sg.Neut}, \text{PART.Zu}, \text{ADJA.Pos.Nom.Sg})
\]
Step 2: The conditioning context is reduced to the most informative attributes.

\[
p(ADJA.Pos.Nom.Sg.Neut \mid ART.Def.Nom.Sg.Neut, PART.Zu) = \\
p(ADJA \mid ART, PART.Zu) \\
p(Pos \mid ART, PART, ADJA) \\
p(Nom \mid ART.Nom, PART, ADJA) \\
p(Sg \mid ART.Sg, PART, ADJA) \\
p(Neut \mid ART.Neut, PART, ADJA) \\
= 1 * 1 * 1 * 1 * 1 = 1 \quad (\text{MLE was 0!})
\]

Question: How to select the most informative context attributes?

Answer: Decision trees
Development of a statistical parser with

- a fine-grained set of grammar symbols such as `NP.base.subj.nom.sg.fem`
- decomposition of the `rule` probabilities into a product of symbol probabilities
- decomposition of the `symbol` probabilities into a product of attribute probabilities
- estimation of the `attribute` probabilities with probability estimation trees
- beam search strategy
Project Goals

- Verb Clustering
  - Extension of the verb-clustering model to deal with *adjuncts*
  - Induction of *noun taxonomies*

- Parsing
  - A statistical parser using *very large contexts* for syntactic disambiguation
  - Selection of the relevant subset of context information with *decision trees*