Predicting Functional Elements in German

Aleksandar Dimitrov

June 13, 2011

Outline

Motivation

Method

Results

Future Work

The Problem

Functional elements exist as syntactic markers without (or with little) semantic value

- Articles (esp. in the case of grammatical gender)
- Prepositions (often mandated by verbs or syntactical structure)

Note: this is mostly a Indo-European (inflectional) view. There might be significant differences for other language families.

- Since they carry no meaning, FE use should be governed by purely distributional properties (similar to allophones in phonetics)

Prior Work

- (De Felice and Pulman 2007) and (De Felice 2008) work on predicting prepositions from given syntactic and lexical contexts using a voted perceptron algorithm
- (Gamon et. al 2008) work on learner language, using a Gigaword n-gram based classifier
- (Tetreault and Chodorw 2008) work on text from language learners using a maximum entropy classifier
- (Bergsma et al. 2009) use Google 5-grams to train an SVM-based classifier

Prior Work

- (De Felice and Pulman 2007) and (De Felice 2008) work on predicting prepositions from given syntactic and lexical contexts using a voted perceptron algorithm
- (Gamon et. al 2008) work on learner language, using a Gigaword n-gram based classifier
- (Tetreault and Chodorw 2008) work on text from language learners using a maximum entropy classifier
- (Bergsma et al. 2009) use Google 5-grams to train an SVM-based classifier
- (Elghafari, Meurers, Wunsch 2010) predict prepositions in normal text using a quantitative web-as-corpus approach
- All work (I know of) has been done for English only

How will we perform on German?

Motivation

- This task sees high accuracy figures across the board for English
- German has a richer morphology, aiding us by providing more contextual information, but also reducing likelihood of recognizable contexts
- German has freer word order, increasing difficulty of this task
- There might be more lexically *different* (and frequently used) prepositions in German than in English
- German has zircumpositions (*aus . . . heraus*) and other funky syntactic constructions

Helping Language Learners

- Preposition and article errors are among the most frequent mistakes in SLA
- Since (ideally) semantics should not play a role, FE prediction should be possible
- Even so, bad quality of learner language will skew distribution-based prediction

Also Helping Language "Knowers"

a.k.a. 'Spell Czech'

- Overall much better quality of surrounding language should aid quantitative methods
- Errors in preposition (Tetreault and Chodorow 2008) and determiner (Turner and Charniak 2007) use are among the most frequent for learners in general, but also among speakers with good command of the language
- Easier than in the case for language learning (fewer other mistakes, longer, well-structured sentences provide context to work with)

Uses in Language Synthesis

- Language synthesis is used to realize machine-represented statements as language
- FE prediction in machine translation: FEs are usually incompatible across languages
- Knowledge engine and semantic representation systems do not compute with syntactic fluff (such as FEs)
- Example: *I went to school yesterday*

  → `go(tense:past-simple`
  `,adv:[yesterday,#non-habitual]`
  `,agent:#speaker,patient:school)`
  →`gehen(tense:präteritum,`
  `adv:[gestern,#non-habitual],`
  `agent:#speaker,patient:Schule)`
  → *Ich ging gestern zur Schule.*

Method

The Data

- German DEWAC corpus (German Web as Corpus)
- Baroni and Kilgarriff 2006, available for multiple languages
- Tree-Tagger tagged, with lemma information
- 24 GB of plain text
- Generally rather noisy, but also very decent, for the size of the corpus

Challenges

- Large corpus data needed to be efficiently parsed and evaluated
- Relatively noisy data needs to be handled robustly
- Since a lot of the ”meat” of the work was due to trial-and-error on huge data sets, efficient choices for implementation and data base backend were necessary

Training

- Training the current system takes 22hrs on recent hardware.
- Resulting data base size is around 12GB.
- Bottlenecks are: hard disk access, RAM availability, data base interaction

The Matching Algorithm

- Direct surface matches are not always possible
- Clever backoff is key to realizing a strong recall

A demonstration

Original item.
Nach 1990 geschah auf Malta fast nichts.

A demonstration

Remove preposition, save as target

Nach 1990 geschah   Malta fast nichts. → auf

A demonstration

Query with all numbers as `CARD`
Nach CARD geschah Malta fast nichts. → auf

A demonstration

Query with all named entities as NE
Nach CARD geschah NE fast nichts. → auf

A demonstration

Query with all finite verbs as lemmas
Nach CARD geschehen NE fast nichts. → auf

A demonstration

Query with far left and far right context as PoS tag

APPR CARD geschehen NE fast PTKNEG. → auf

A demonstration

Query with middle left and middle right context as PoS tag
APPR CARD geschehen   NE ADV PTKNEG. → auf

A demonstration

Query with close left and close right context as PoS tag
APPR CARD VVFIN   NE ADV PTKNEG. → auf

A demonstration

APPR CARD VVFIN   NE ADV PTKNEG. → auf

A demonstration

Backoff
1990 geschah Malta fast → auf

A demonstration

Query with all numbers as `CARD`
CARD geschah Malta fast → auf

A demonstration

Query with all named entities as `NE`
CARD geschah NE fast → auf

A demonstration

Query with all finite verbs as lemmas
CARD geschehen NE fast → auf

A demonstration

Query with far left and far right context as PoS tag
CARD geschehen   NE ADV → auf

A demonstration

Query with close left and close right context as PoS tag
CARD VVFIN   NE ADV → auf

A demonstration

Backoff
geschah   Malta → auf

A demonstration

Query with all numbers as `CARD`
geschah   Malta → auf

A demonstration

Query with all named entities as `NE`

geschah NE → auf

A demonstration

Query with all finite verbs as lemmas
geschehen NE → auf

A demonstration

Query with left and right context as PoS tag
VVFIN   NE → auf

A demonstration

VVFIN NE → auf

Majority baseline: in

A demonstration

VVFIN   NE → auf

Results

- Prohibitive training and evaluation costs make analysis and incremental improvement a day-long endeavour
- Preliminary results on total accuracy, precision and recall are available
- More detailed analysis data for recall & precision during every step of the matching algorithm is on the way

What is Performance, Anyway?

- The ultimate goal is predicting prepositions in the chosen set correctly
  Precision: How many did were predicted correctly?
- But we want to do so based on prior evidence, not wild guessing
  Recall: How many times were we able to find an item in the database (in any form?)
- Backoff will tend to increase overall recall, at the cost of precision
  (This is as it should be.)
- Graphed figures of precision and recall of the whole algorithm yield effectiveness estimates of every step (cumulatively, or separately.)

Future Work

- Current db-based design is slow to train
- Expensive to run or redistribute
- Limited to predicting a handful of prepositions
- Depends on shallow linguistic processing and *huge* amounts of input data to do well
- Isn't particularly smart or ingenuous

Using Machine Learning

- The context database is around 50% the size of the original corpus
- Prohibitive lookup costs and portability
- Lossy compression could be used to increase viability for particular purposes
- N-Gram based Language models, in particular `SLIRM` or `BerkeleyLM` seem a good fit
- How well would a memory based learner perform (k-nearest neighbour seems similar to the matching algorithm?)
- Other methods? (HMM guided automata, maximum entropy language models, . . .)

Predicting Sets of Prepositions Instead

- Frequently, items that are also FEs *do* carry *some* meaning
- Indiscernible when looking at the surface form
- Sometimes, multiple FEs are admissible and it is not possible to make a distinction (either a lot of linguistic or world knowledge is required, or there isn't a distinction)
- Example (semantic meaning:) *Der Hund war* {*neben,unter,über,auf,. . .*} *dem Tisch*
- For grammatic correction of SLLs or even native speakers, predicting *sets* of admissible FEs in a given context might be enough (or even much more desirable)

Automatic Detection of Collapsible Classes

- `CARD`, `NE`, lemmas, etc. are arbitrarily chosen
- Algorithms exist to detect possible classes of local variability in large data sets
  1. Change one thing in a context item
  2. How many cases exist with only this one item changed?
  3. On average (and in particular,) how do the distributional properties of those cases compare?
- Conceptually easy to implement, but
- Have to choose sensible thresholds or estimate intelligently
- Operation as a whole would be very slow

Beyond Words: Functional Elements in Syntax and Morphology

The general case of a functional element is too general, and unlikely to be solved only once, with one method.

- Morphology: Case markings, agglutinated prepositions or articles
- Syntax: Word order