Assigning deep lexical types in Portuguese and English

João Silva

Faculty of Sciences, University of Lisbon
NLX — Natural Language and Speech Group

9th DELPH-IN Summit
St. Wendel, Germany
July 2013

Presentation outline

1. Introduction and recap
2. The experiment
3. Closing remarks

Presentation outline

Motivation/Approach

Assigning deep lexical types to unknown words

- LX-Gram, an HPSG for Portuguese
- Generics for unknown word handling
  shallow pre-processing using LX-Suite
  part-of-speech $\rightarrow$ deep type

Approach

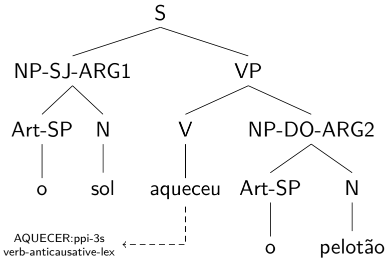
- On-the-fly pre-processing
- Structured features
  e.g. syntactic constituency, grammatical dependencies, etc.
- Off-the-shelf tools

On a previous DELPH-IN Summit. . .

Vista extraction

lkb2standard

- Runs over data exported by tsdb
- Normalization: X-bar, punctuation, empty nodes, slashes, . . .
- Add information to leafs: Lemma, inflection, lexical type, . . .
- Other fixes



the sun warmed the platoon

To see more, check the Treebank Searcher at:
http://lxcenter.di.fc.ul.pt

On a previous DELPH-IN Summit...

SVM and tree kernels

Support-vector machines

- Machine-learning, linear binary classifier
- Instances as vectors in $\mathbb{R}^n$, dot product measures similarity

Representing structure as feature vectors

- Kernel trick, convolution kernels
- For trees: Number of subtrees in common between two trees

Software

- Tree kernel by Alessandro Moschitti (SVM-TK)
- SVM by Thorsten Joachims (SVM-Light)

On a previous DELPH-IN Summit…

The SVM-TK classifier

SVM is a binary classifier

- One-vs-one voting strategy
  - One classifier for each pair of types
    i.e. $\frac{n \cdot (n-1)}{2}$ classifiers
  - Choose the type that got the most votes

Data-sparseness

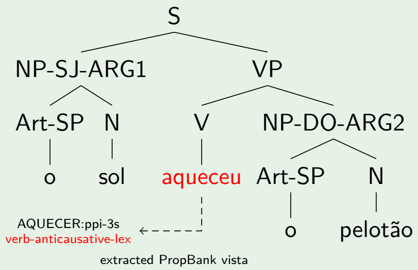
- Restrict to top-*n* (most frequent) types in a category
- Focus mostly on verbal types

But how is “structure” encoded?

On a previous DELPH-IN Summit...

The SVM-TK classifier: Encoding "structure" in features

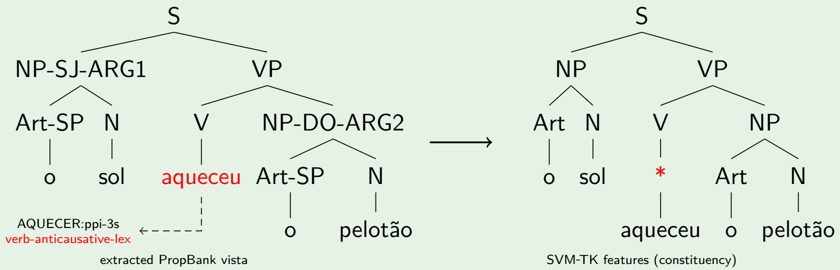
A positive instance of the verb-anticausative-lex type



On a previous DELPH-IN Summit...

The SVM-TK classifier: Encoding "structure" in features

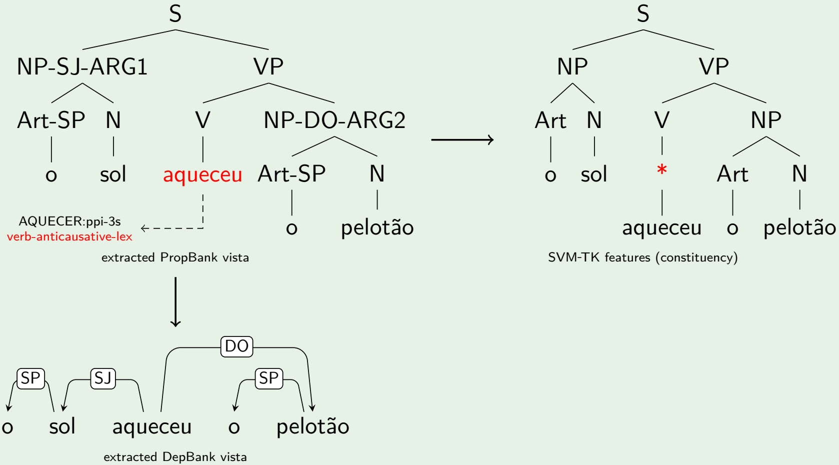
A positive instance of the `verb-anticausative-lex` type



On a previous DELPH-IN Summit...

The SVM-TK classifier: Encoding "structure" in features

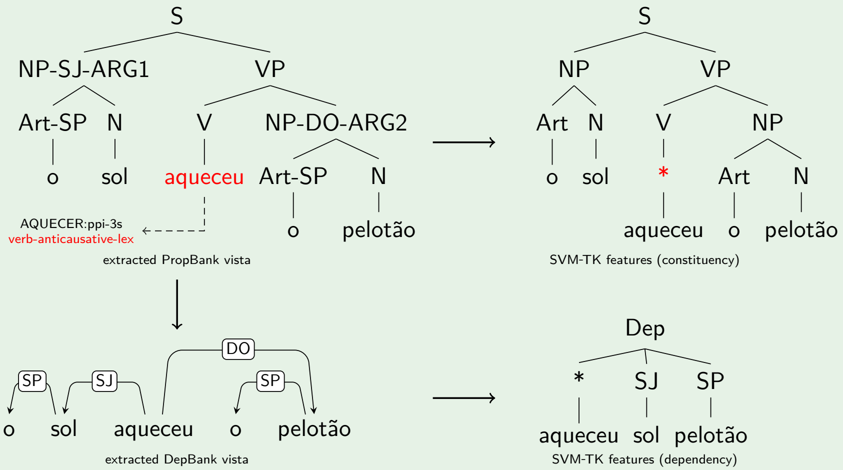
A positive instance of the `verb-anticausative-lex` type



On a previous DELPH-IN Summit. . .

The SVM-TK classifier: Encoding “structure” in features

A positive instance of the `verb-anticausative-lex` type



On a previous DELPH-IN Summit...

Early experiments

Setup

- DeepGramBank: $5,422$ sentences, 130 verb types
- PropBank, TreeBank and DepBank vistas (gold data)
- Over top-10 verb types
- 10-fold cross-validation
- Comparison with TnT POS-tagger

Results

- Dependency features were best, slightly above TnT
  ($92.28\% > 92.16\%$)

Since then…

- Expand the set of assignable types
  - Top-10, top-20, top-30, …
    (verb token coverage: 68%, 84%, 90%, …)
  - Data-sparseness makes assigning from the full set unfeasible
  - SVM-TK loses to TnT as *n* increases
- Use predicted dependencies
  - MaltParser, running at 88% LAS
  - Slight detrimental impact
    NB: Training over predicted data helps

Since then...

- Test on extended datasets (automatically annotated)
  - Run LX-Gram, take the top-ranked analysis
  - Progressively larger datasets: 5k $\rightarrow$ 10k $\rightarrow$ 15k $\rightarrow$ 20k
  - On the largest dataset, SVM-TK beats TnT
    (even on top-30 with predicted features)
- Compare with in-grammar disambiguation
  - Allow unknown word to have *n* types, let LX-Gram disambiguate
  - In-grammar disambiguation performs worse
- Run on ERG/Redwoods

Presentation outline

Running on ERG/Redwoods

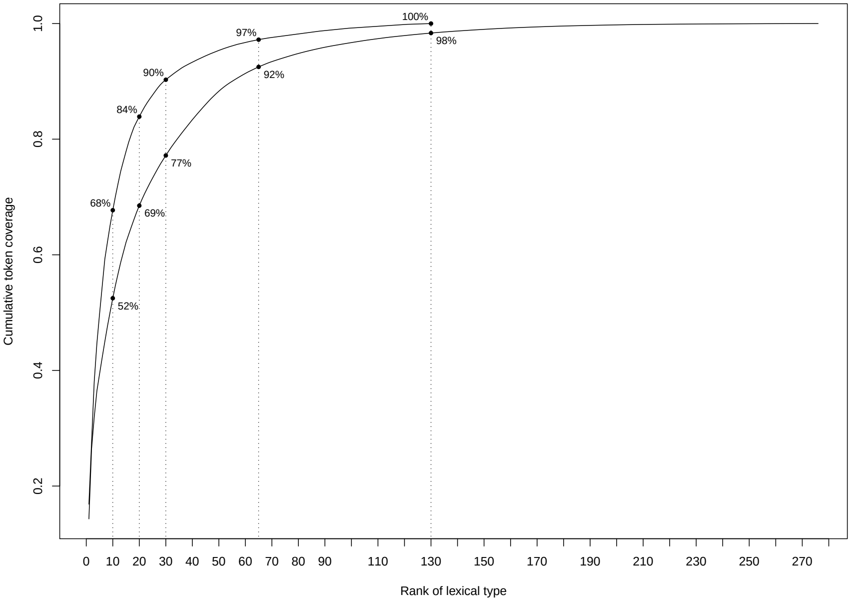
The corpus

- Obtaining CoNLL from Redwoods
  (thanks to Angelina Ivanova for helping with this)
- Close to 45k sentences, 276 verb types
  $\frac{276}{130} \approx 2.12$ times as many as in DeepGramBank

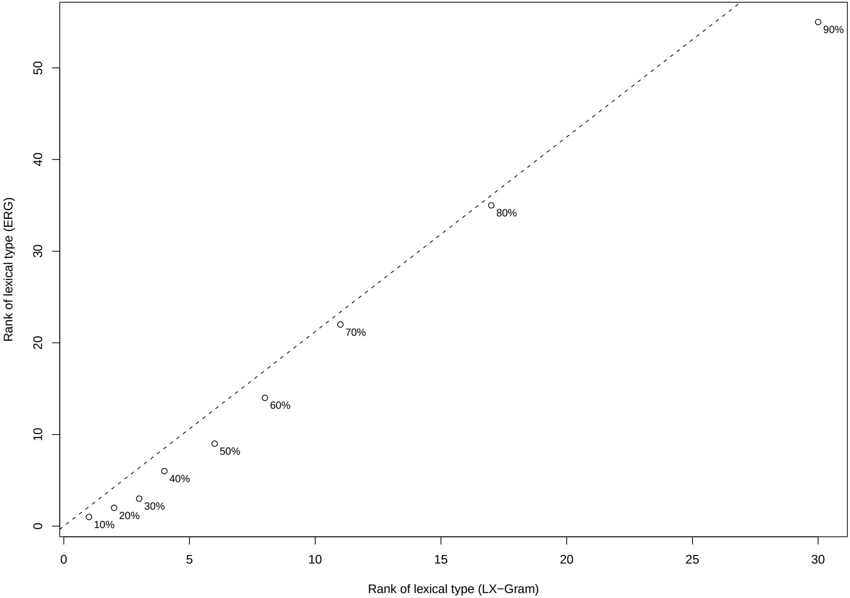
Setup

- SVM-TK classifier
  grammatical dependencies as features
- 10-fold cross-validation
- Top-*n* verbs

Verb token coverage (given *n*-th rank)



Verb *n*-th rank coverage correspondence



Results

Comparison with TnT, over top-*n* verb types (%)

	SVM-TK	TnT
top-10	94.76	92.96
top-20	90.27	91.69
top-30	89.04	91.62

LX-Gram/DeepGramBank

	SVM-TK	TnT
top-19	93.05	89.49
top-41	91.63	87.82
top-56	90.93	87.26

ERG/Redwoods

- SVM-TK consistently outperforms TnT
  (given enough training data)

Presentation outline

Closing remarks

In a nutshell

The goal

- Combine strengths: deep analysis + robust parsing
  (automatically assigning lexical types to unknown words)

The way

- Off-the-shelf tools
- SVM-TK classifier that takes dependencies as features

The result

- Improves on current approach
  (but requires more data)

Thank you.