Assigning lexical types using tree kernels

João Ricardo Martins Ferreira da Silva
Supervisor: Prof. Dr. António Branco

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

DELPH-IN Summit
Suquamish, WA
June 2011

Presentation outline

Presentation outline

Task and approach

Assigning lexical types to (unknown) words

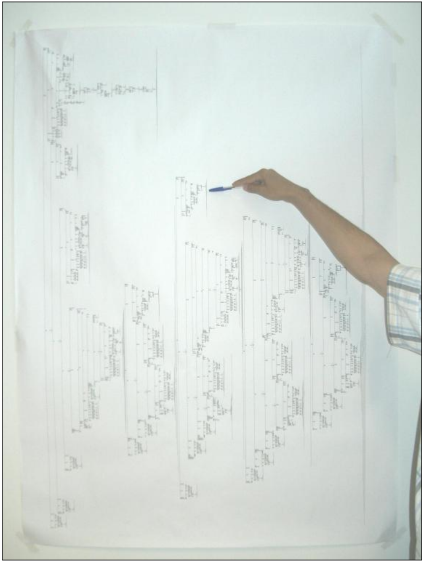
- On-line/on-the-fly
- Machine learning
- Use structured information
  (viz. constituency and/or dependency)

Presentation outline

DeepGramBank

Overview

- Produced with LX-Gram
  (two annotators + adjudicator)
- 5,422 sentences in Portuguese
  (mostly newspaper excerpts)
- Breakdown of some types:
  129 types of verb
  146 types of noun
  75 types of adjective
  (each is highly skewed)



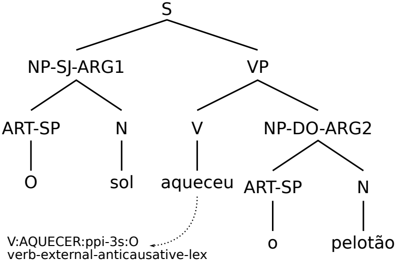
Representation of "Todos os computadores têm um disco"
(arm provided for scale)

Extracting vistas

DeepGramBank to PropBank

lkb2standard

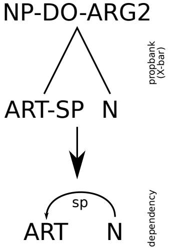
- Runs over data exported by tsdb
- Normalization: X-bar, punctuation, empty nodes, slashes, ...
- Add information to leafs: Lemma, inflection, lexical type, ...
- Other fixes



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

propbank2dependency

- Runs over the PropBank
- Output:
  - Dependency triples and/or CoNLL format
- NB: not implemented by me



Dependency extraction (general rule)

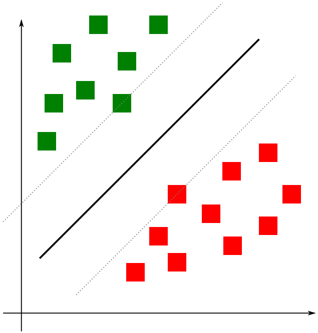
Presentation outline

1. Introduction
2. Dataset
3. **Classifiers**
4. Conclusion

Support vector machine (SVM)

Overview

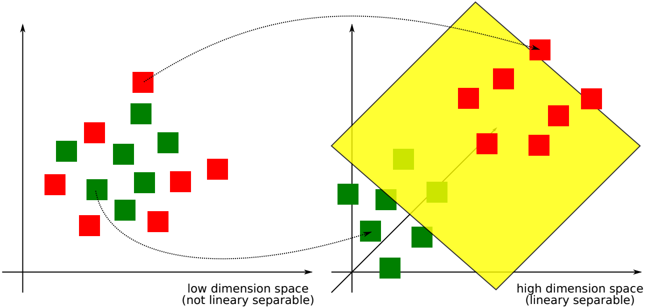
- Machine-learning algorithm
- Linear binary classifier
- Instances as vectors in $\mathbb{R}^n$
- Learns separating hyperplane
  (with maximal margin)
- Uses the dot product
  (to measure vector similarity)



Support vector machine (SVM)

Kernel trick

- Kernel function replaces dot product
  (if certain conditions are met)
- Feature vectors are not explicitly generated



SVM with tree kernels (SVM-TK)

Representing parse trees as feature vectors

- Kernel function:
  Number of subtrees in common between two trees

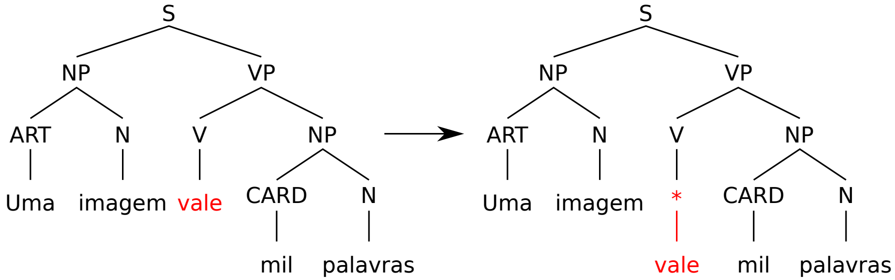
Software

- Tree kernel by Alessandro Moschitti (SVM-TK)
  `http://disi.unitn.it/moschitti/Tree-Kernel.htm`
- SVM by Thorsten Joachims (SVM-Light)
  `http://svmlight.joachims.org`

Features for tree kernel

TreeBank/PropBank

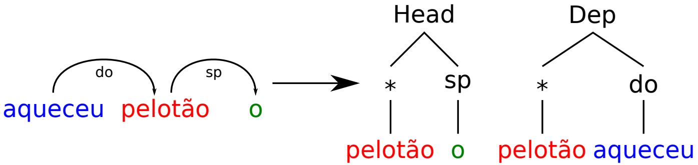
Take whole parse tree, mark pre-terminal node of target



Features for tree kernel

DepBank

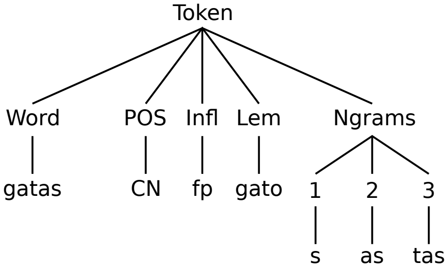
Take immediate dependents and head of target



NB: In SVM-TK, an instance can have multiple vectors and tress

Features for tree kernel
Token

Word form, POS, lemma, n-grams, etc.



OvA and OvO

Voting strategy?

One-vs-all (OvA)

- One per lexical type
  i.e. n classifiers
- Choose best vote

One-vs-one (OvO)

- One per each pair of types
  i.e. $n \cdot (n-1)/2$ classifiers
- Choose most voted

Evaluation

10-fold cross-evaluation for top-10 verbal types
(3188 instances: $1107, 434, 333, \ldots, 140$)

- Sequential classifiers

TnT	92.16%

- Pointwise classifiers
  (SVM-TK, OvO voting, over gold data)

treebank	86.98%
propbank	89.05%
depbank	92.28%

Presentation outline

Closing remarks

- Other experiments that did not fare well:
  C&C, TiMBL, SMOTE, ...
- Trees for SVM-TK are a very flexible approach
  (one can encode a great variety of features)
- SVM-TK vs. TnT
  (results shown for SVM-TK are over gold data)
- Unbalanced classes
- Data sparseness issues

Thank you.