# Natural Language Processing and Information Retrieval

## State-of-the-art Kernels in Natural Language Processing

**Alessandro Moschitti**

Dept. of Computer Science and Engineering

University of Trento

moschitti@disi.unitn.it

# Outline: preliminaries

- Motivation
- Structural Kernels
  - Semantic/Syntactic Tree Kernels
    - PTK
    - SPTK
- Kernels for question answering
  - Question Classification
  - Jeopardy Cue Classification
  - Answer reranking

# Outline: Kernels for NLP applications

- NLP applications
  - Semantic Role Labeling
  - Relation Extraction
  - Coreference Resolution
  - Textual Entailment Recognition
- Kernels for Reranking
  - Spoken Language Understanding
  - Named Entity Recognition

# Motivation (1)

- Feature design most difficult aspect in designing a learning system
  - complex and difficult phase, e.g., structural feature representation:
  - deep knowledge and intuitions are required
  - design problems when the phenomenon is described by many features

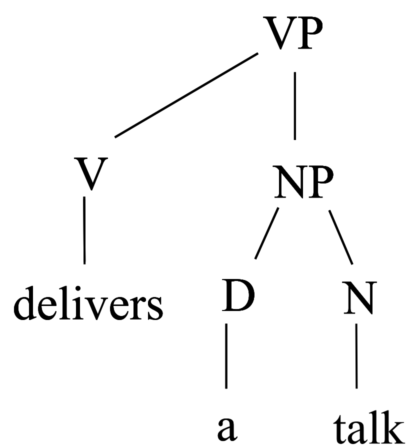
# Motivation (2)

- Kernel methods alleviate such problems
  - Structures represented in terms of substructures
  - High dimensional feature spaces
  - Implicit and abstract feature spaces
- Generate high number of features
  - Support Vector Machines “select” the relevant features
  - Automatic feature engineering side-effect

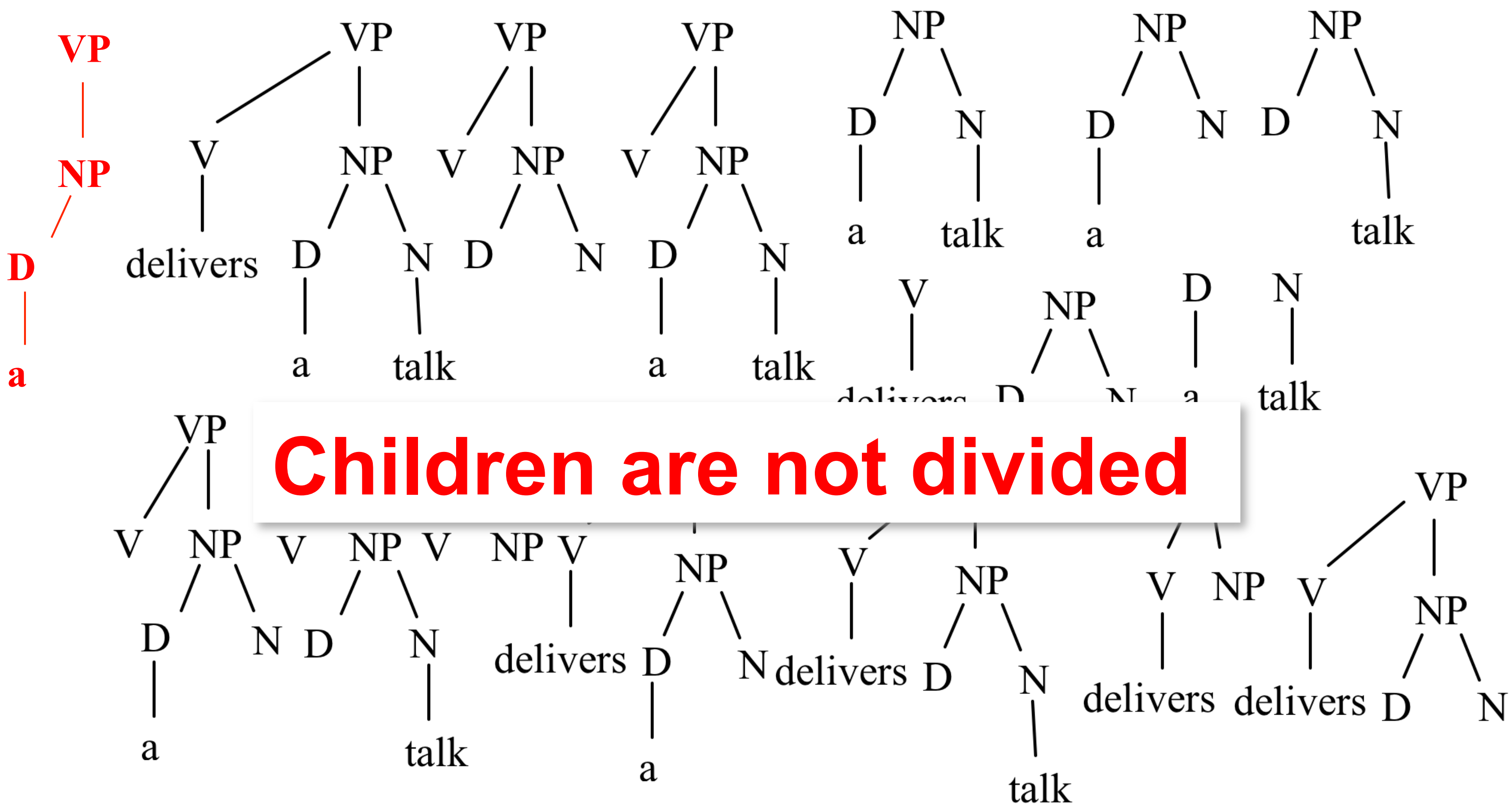
# Motivation (3)

- High accuracy especially for new applications and new domains
  - Manual engineering still poor, e.g. arabic SRL
- Inherent higher accuracy when many structural patterns are needed, e.g. Relation Extraction
- Fast prototyping and adaptation for new domains and applications

# The Syntactic Tree Kernel (STK)
## [Collins and Duffy, 2002]



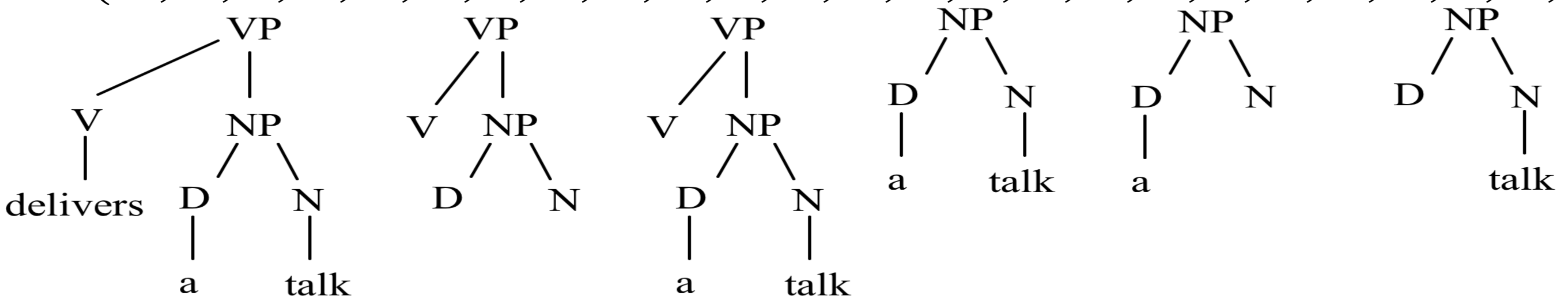
# The overall fragment set



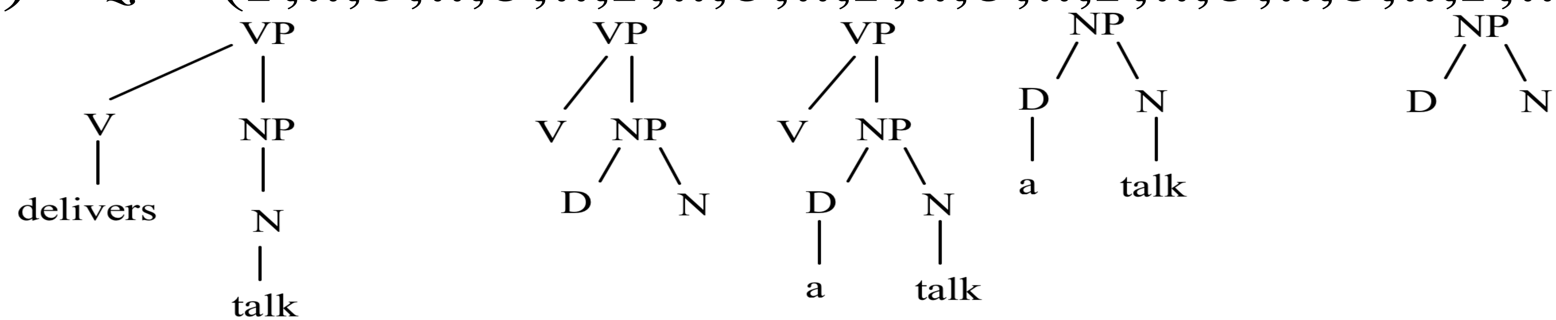

Children are not divided

# Explicit kernel space

$$\phi(T_x) = \vec{x} = (0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0)$$




$$\phi(T_z) = \vec{z} = (1,..,0,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,0,..,1,..,0,..,0)$$




- $\vec{x} \cdot \vec{z}$ counts the number of common substructures

# Efficient evaluation of the scalar product

$$\vec{x} \cdot \vec{z} = \phi(T_x) \cdot \phi(T_z) = K(T_x, T_z) =$$

$$= \sum_{n_x \in T_x} \sum_{n_z \in T_z} \Delta(n_x, n_z)$$

# Efficient evaluation of the scalar product

$$\vec{x} \cdot \vec{z} = \phi(T_x) \cdot \phi(T_z) = K(T_x, T_z) =$$

$$= \sum_{n_x \in T_x} \sum_{n_z \in T_z} \Delta(n_x, n_z)$$

- [Collins and Duffy, ACL 2002] evaluate Δ in O(n²):

$$\Delta(n_x, n_z) = 0, \quad \text{if the productions are different else}$$

$$\Delta(n_x, n_z) = 1, \quad \text{if pre-terminals else}$$

$$\Delta(n_x, n_z) = \prod_{j=1}^{nc(n_x)} (1 + \Delta(ch(n_x, j), ch(n_z, j)))$$

# Other Adjustments

- Decay factor

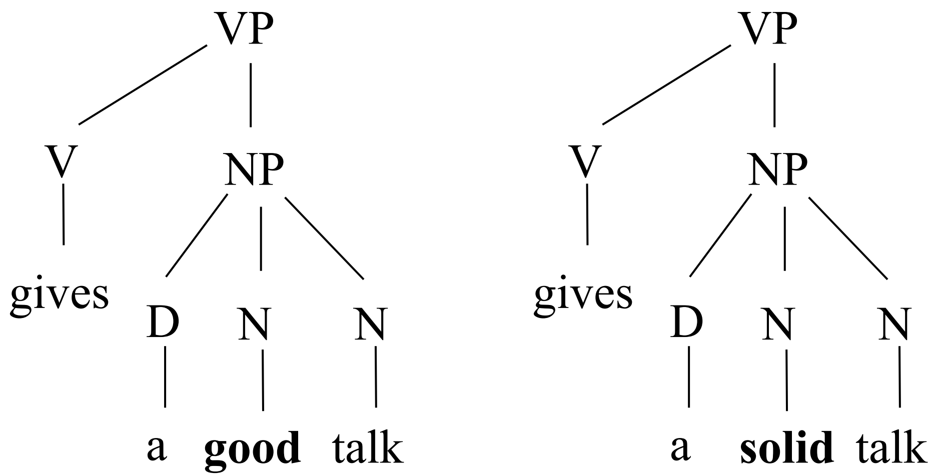
$$\Delta(n_x, n_z) = \lambda, \quad \text{if pre-terminals else}$$

$$\Delta(n_x, n_z) = \lambda \prod_{j=1}^{nc(n_x)} (1 + \Delta(ch(n_x, j), ch(n_z, j)))$$

- Normalization

$$K'(T_x, T_z) = \frac{K(T_x, T_z)}{\sqrt{K(T_x, T_x) \times K(T_z, T_z)}}$$

# Syntactic/Semantic Tree Kernels
## [Bloehdorn & Moschitti, ECIR 2007 & CIKM 2007]




- Similarity between the fragment leaves
  - Tree kernels + Lexical Similarity Kernel

# Syntactic/Semantic Tree Kernels
## [Bloehdorn & Moschitti, ECIR 2007 & CIKM 2007]

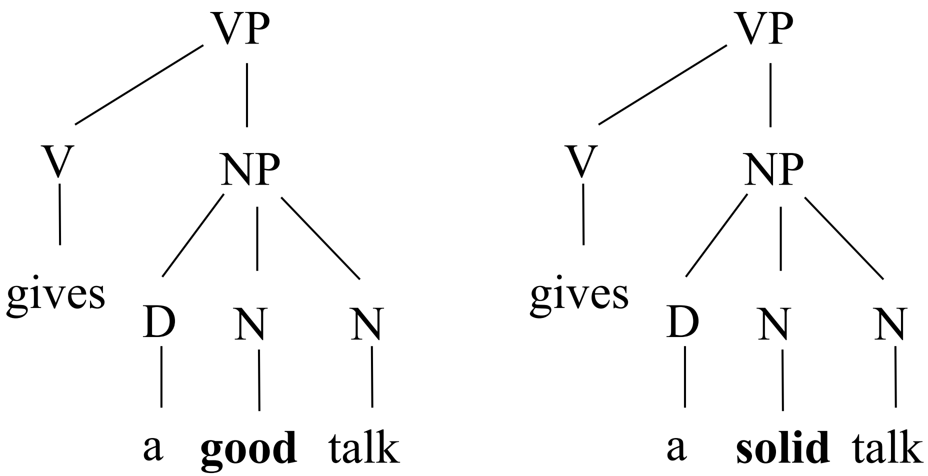
**Definition 4 (Tree Fragment Similarity Kernel).** *For two tree fragments $f_1, f_2 \in \mathcal{F}$, we define the Tree Fragment Similarity Kernel as*[4]*:*

$$\kappa_{\mathcal{F}}(f_1, f_2) = comp(f_1, f_2) \prod_{t=1}^{nt(f_1)} \kappa_S(f_1(t), f_2(t))$$

$$\kappa_T(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$$

*where* $\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{|\mathcal{F}|} I_i(n_1) I_j(n_2) \kappa_{\mathcal{F}}(f_i, f_j)$.

# Merging of Kernels




$K_S$(gives,gives)*$K_S$(a,a)*
$K_S$(good,solid)*$K_S$(talk,talk)
= 1 * 1 * 0.5 * 1 = 0.5

$$\kappa_T(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$$

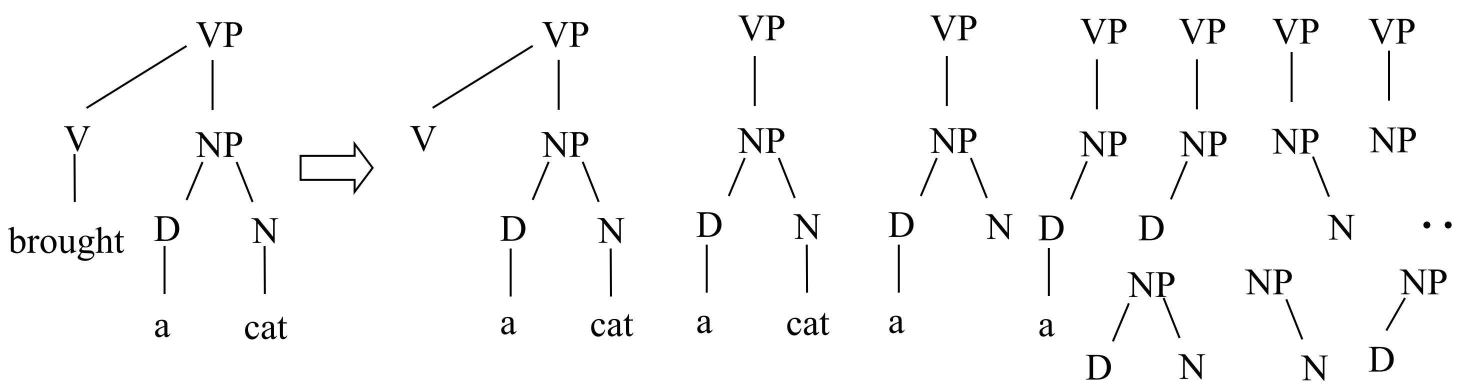
$$where\ \Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \sum_{j=1}^{|\mathcal{F}|} I_i(n_1) I_j(n_2) \kappa_{\mathcal{F}}(f_i, f_j).$$

# Delta Evaluation is very simple

0. if $n_1$ and $n_2$ are pre-terminals and $label(n_1) = label(n_2)$ then $\Delta(n_1, n_2) = \lambda \kappa_S(ch^1_{n_1}, ch^1_{n_2})$,
1. if the productions at $n_1$ and $n_2$ are different then $\Delta(n_1, n_2) = 0$;
2. $\Delta(n_1, n_2) = \lambda$,
3. $\Delta(n_1, n_2) = \lambda \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch^j_{n_1}, ch^j_{n_2}))$.

# Partial Trees, [Moschitti, ECML 2006]

- STK + String Kernel with weighted gaps on Nodes' children



# Partial Tree Kernel

- if the node labels of $n_1$ and $n_2$ are different then $\Delta(n_1, n_2) = 0$;

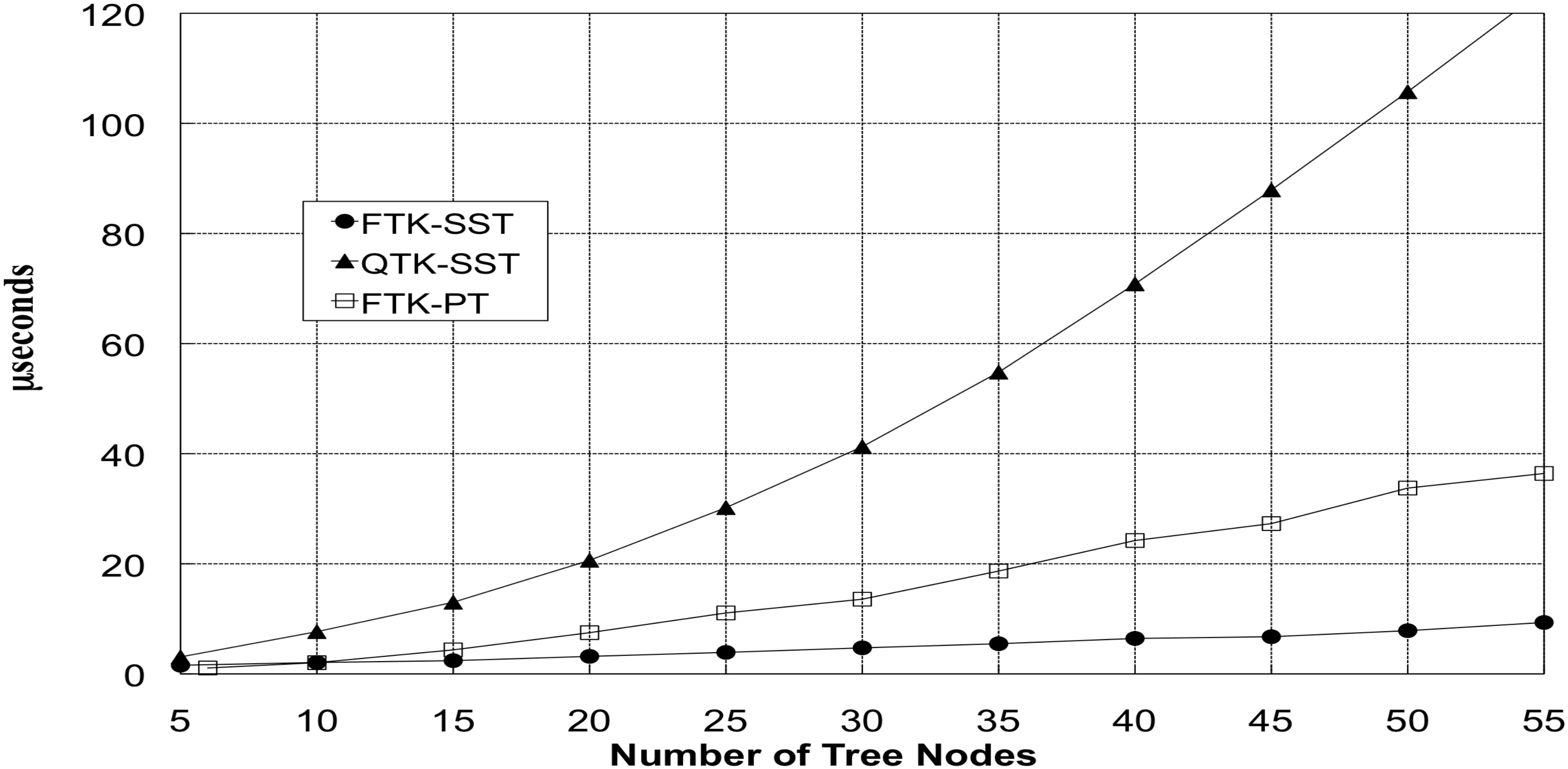
- else

$$\Delta(n_1, n_2) = 1 + \sum_{\vec{J}_1, \vec{J}_2, l(\vec{J}_1) = l(\vec{J}_2)} \prod_{i=1}^{l(\vec{J}_1)} \Delta(c_{n_1}[\vec{J}_{1i}], c_{n_2}[\vec{J}_{2i}])$$

- By adding two decay factors we obtain:

$$\mu\Big(\lambda^2 + \sum_{\vec{J}_1, \vec{J}_2, l(\vec{J}_1) = l(\vec{J}_2)} \lambda^{d(\vec{J}_1) + d(\vec{J}_2)} \prod_{i=1}^{l(\vec{J}_1)} \Delta(c_{n_1}[\vec{J}_{1i}], c_{n_2}[\vec{J}_{2i}])\Big)$$

# Running Time of Tree Kernel Functions

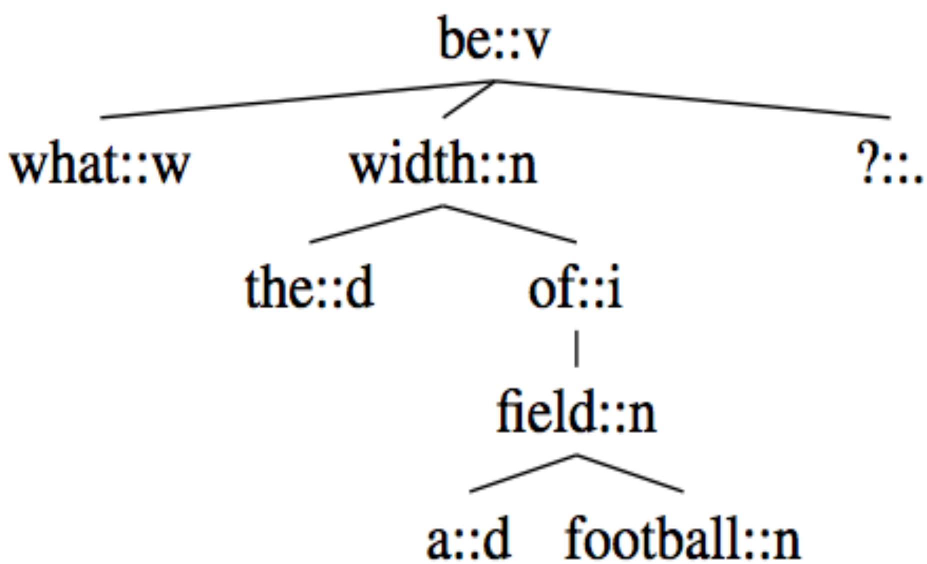


# Smoothed Partial Tree Kernels

- Same idea of Syntactic Semantic Tree Kernel but the similarity is extended to any node of the tree
- The tree fragments are those generated by PTK
- Basically it extends PTK with similarities

# Examples of Dependency Trees

- What is the width of a football field?




- SPTK can match with the length of the biggest tennis-court → (length (the) ((the) (biggest (the)(tennis court)))

- Word+generralized POS-tag

# Equation of SPTK

If $n_1$ and $n_2$ are leaves then $\Delta_\sigma(n_1, n_2) = \boxed{\mu\lambda\sigma(n_1, n_2)}$

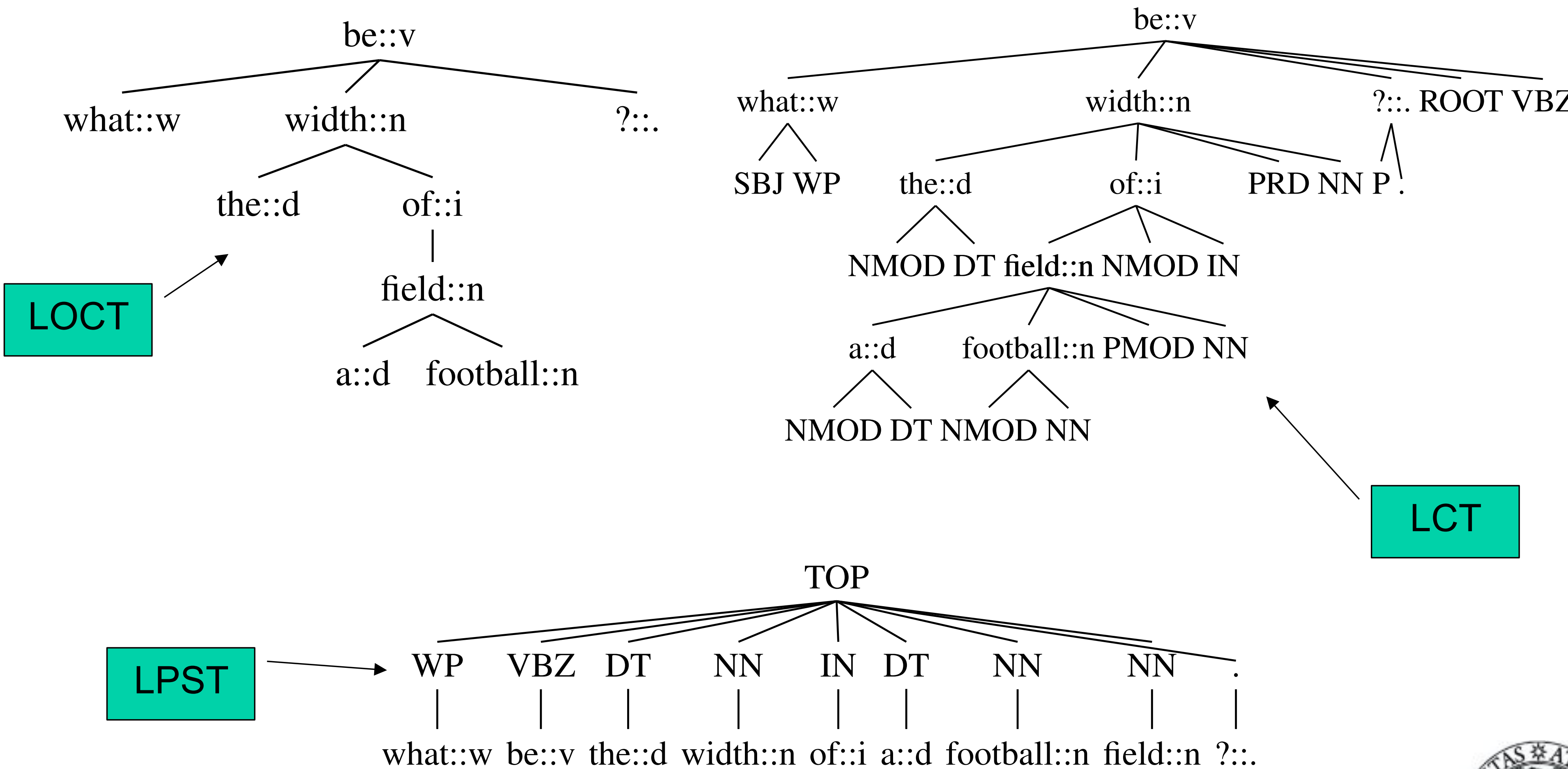
else

$$\Delta_\sigma(n_1, n_2) = \boxed{\mu\sigma(n_1, n_2)} \times \boxed{\left(\lambda^2 + \sum_{\vec{I}_1, \vec{I}_2, l(\vec{I}_1) = l(\vec{I}_2)} \lambda^{d(\vec{I}_1) + d(\vec{I}_2)} \prod_{j=1}^{l(\vec{I}_1)} \Delta_\sigma(c_{n_1}(\vec{I}_{1j}), c_{n_2}(\vec{I}_{2j}))\right)}$$

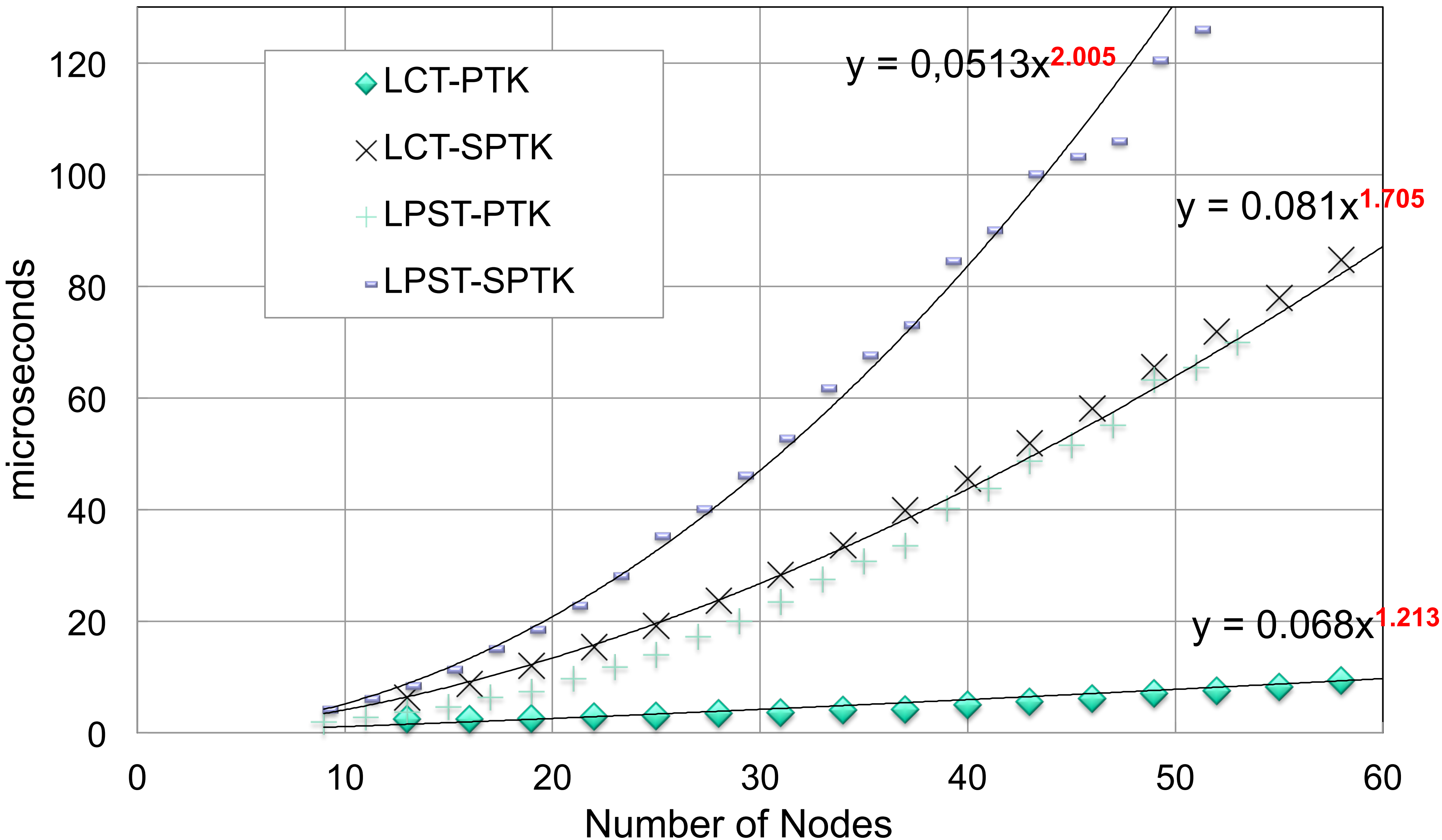
**Lexical Similarity**

**PTK**

# Same Task with PTK, SPTK and Dependency Trees



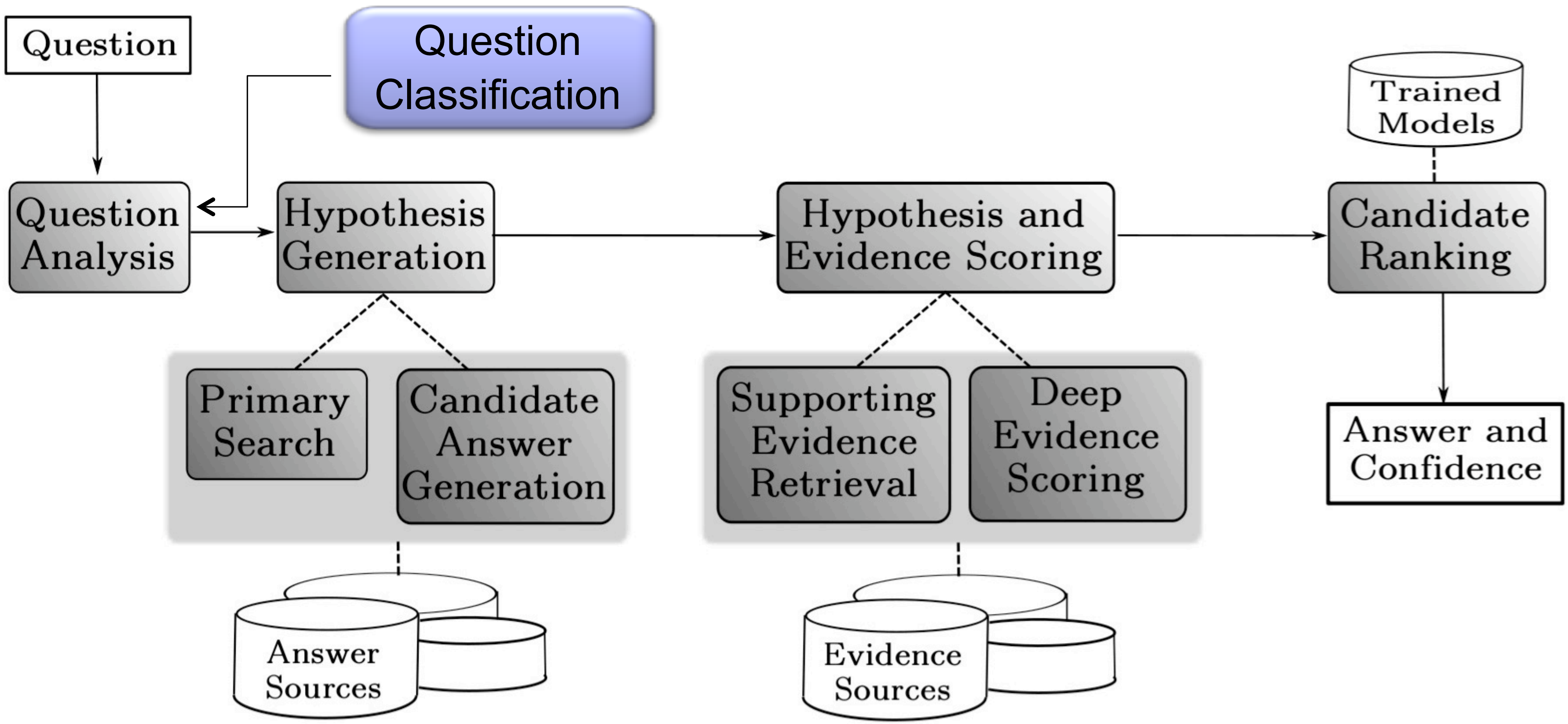
# Tree Kernel Efficiency



# Kernel Methods for Practical Applications

# Question Answering

# A QA Pipeline: Watson Overview



# Question Classification

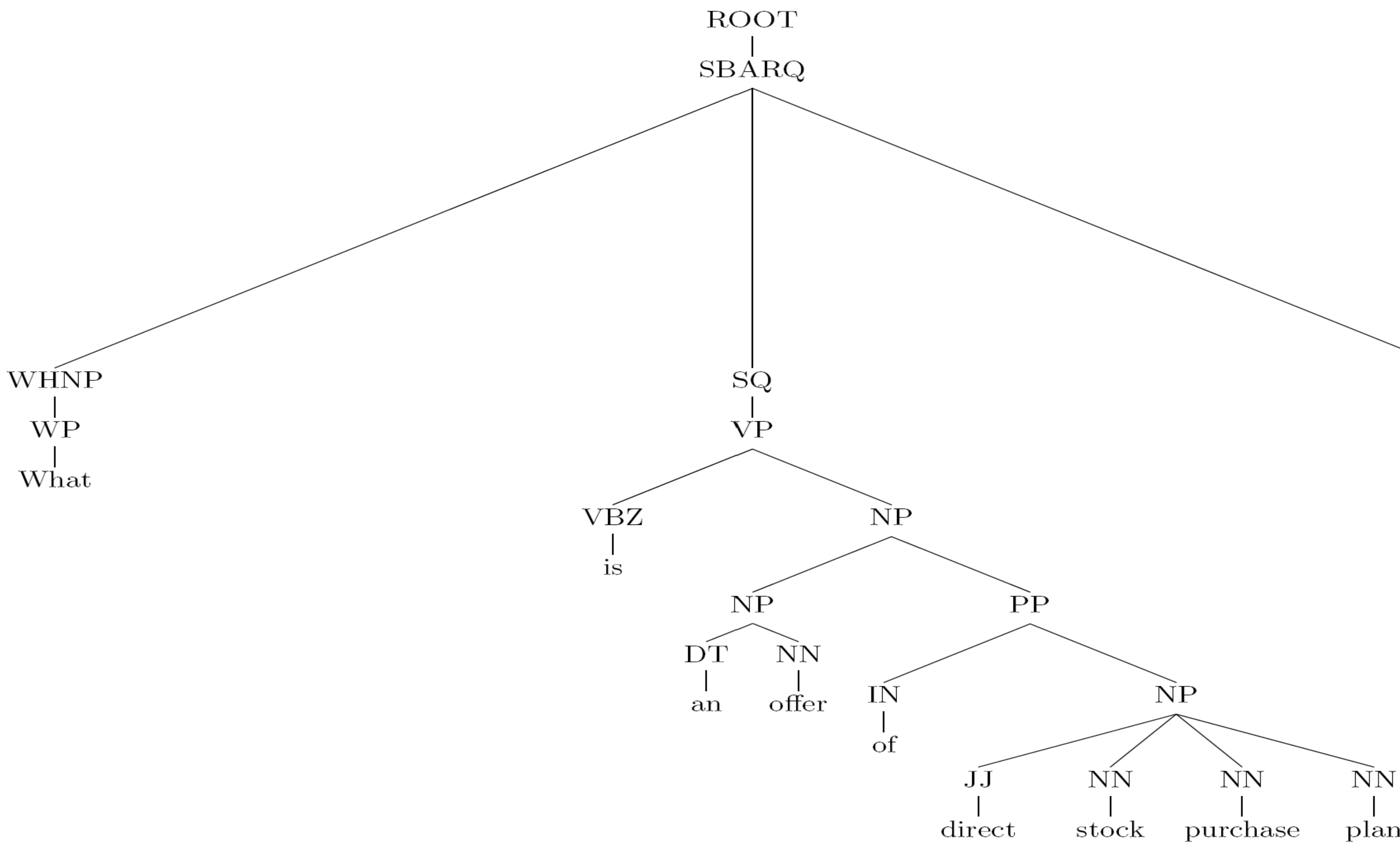
- **Definition**: What does HTML stand for?
- **Description**: What's the final line in the Edgar Allan Poe poem "The Raven"?
- **Entity**: What foods can cause allergic reaction in people?
- **Human**: Who won the Nobel Peace Prize in 1992?
- **Location**: Where is the Statue of Liberty?
- **Manner**: How did Bob Marley die?
- **Numeric**: When was Martin Luther King Jr. born?
- **Organization**: What company makes Bentley cars?

# Question Classifier based on Tree Kernels

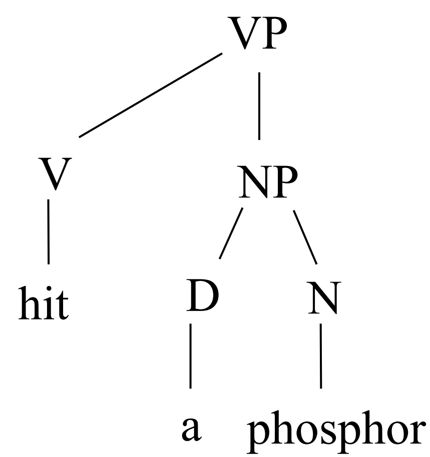
- Question dataset (http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC/) [Lin and Roth, 2005])
  - Distributed on 6 categories: Abbreviations, Descriptions, Entity, Human, Location, and Numeric.
- Fixed split 5500 training and 500 test questions
- Using the whole question parse trees
  - Constituent parsing
  - Example

**"What is an offer of direct stock purchase plan ?"**

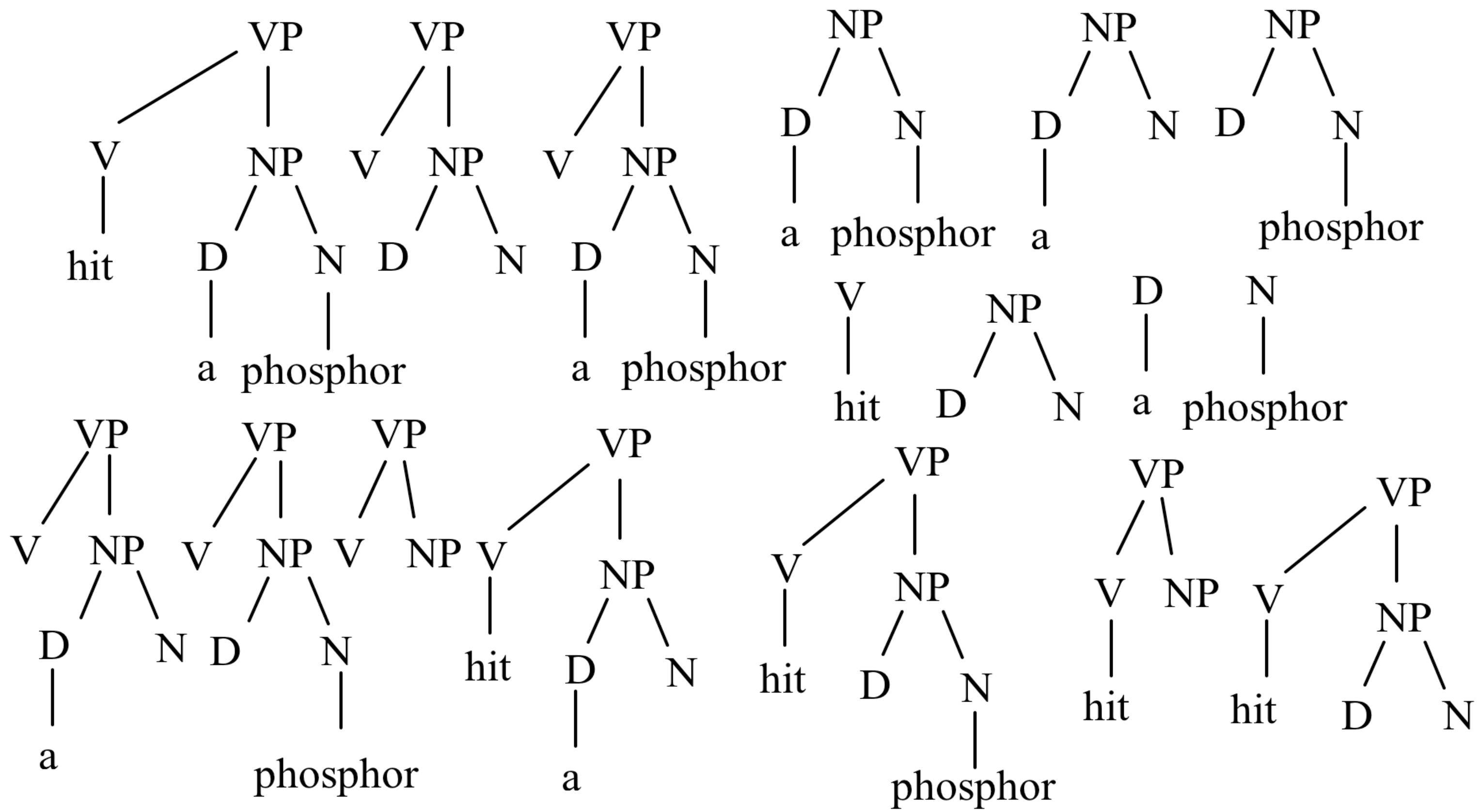
# Syntactic Parse Trees (PT)



# Similarity based on the number of common substructures



# A portion of the substructure set



# Exercise with SVM-light-TK Software

- Encodes ST, STK and combination kernels in SVM-light [Joachims, 1999]
- Available at http://dit.unitn.it/~moschitt/
- Tree forests, vector sets
- The new SVM-Light-TK toolkit will be released asap (email me to have the current version)

# WordNet Hierarchy

## WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: car  Search WordNet

Display Options: (Select option to change)  Change

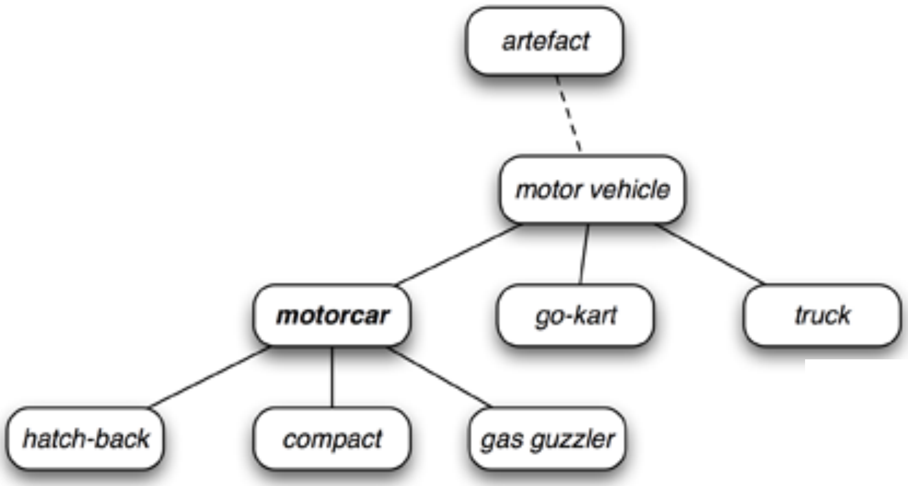
Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

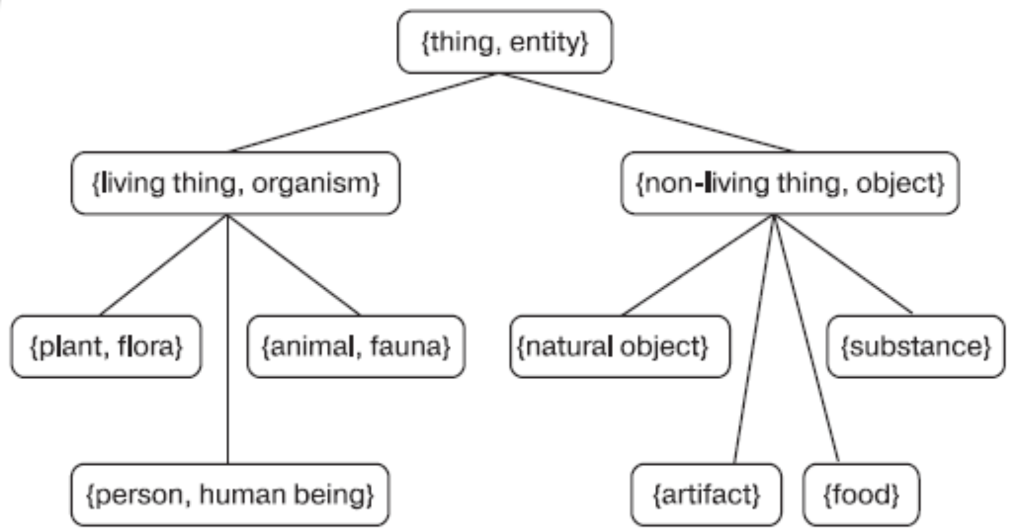
Display options for sense: "an example sentence"

### Noun

- S: (n) **car**, auto, automobile, machine, motorcar *"he needs a car to get to work"*
- S: (n) **car**, railcar, railway car, railroad car *"three cars had jumped the rails"*
- S: (n) **car**, gondola
- S: (n) **car**, elevator car *"the car was on the top floor"*
- S: (n) cable car, **car** *"they took a cable car to the top of the mountain"*

# Sub-hierarchies in WordNet



# Similarity based on WordNet

Inverted Path Length:

$$sim_{IPL}(c_1, c_2) = \frac{1}{(1 + d(c_1, c_2))^{\alpha}}$$

Wu & Palmer:

$$sim_{WUP}(c_1, c_2) = \frac{2\, dep(lso(c_1, c_2))}{d(c_1, lso(c_1, c_2)) + d(c_2, lso(c_1, c_2)) + 2\, dep(lso(c_1, c_2))}$$

Resnik:

$$sim_{RES}(c_1, c_2) = -\log\, P(lso(c_1, c_2))$$

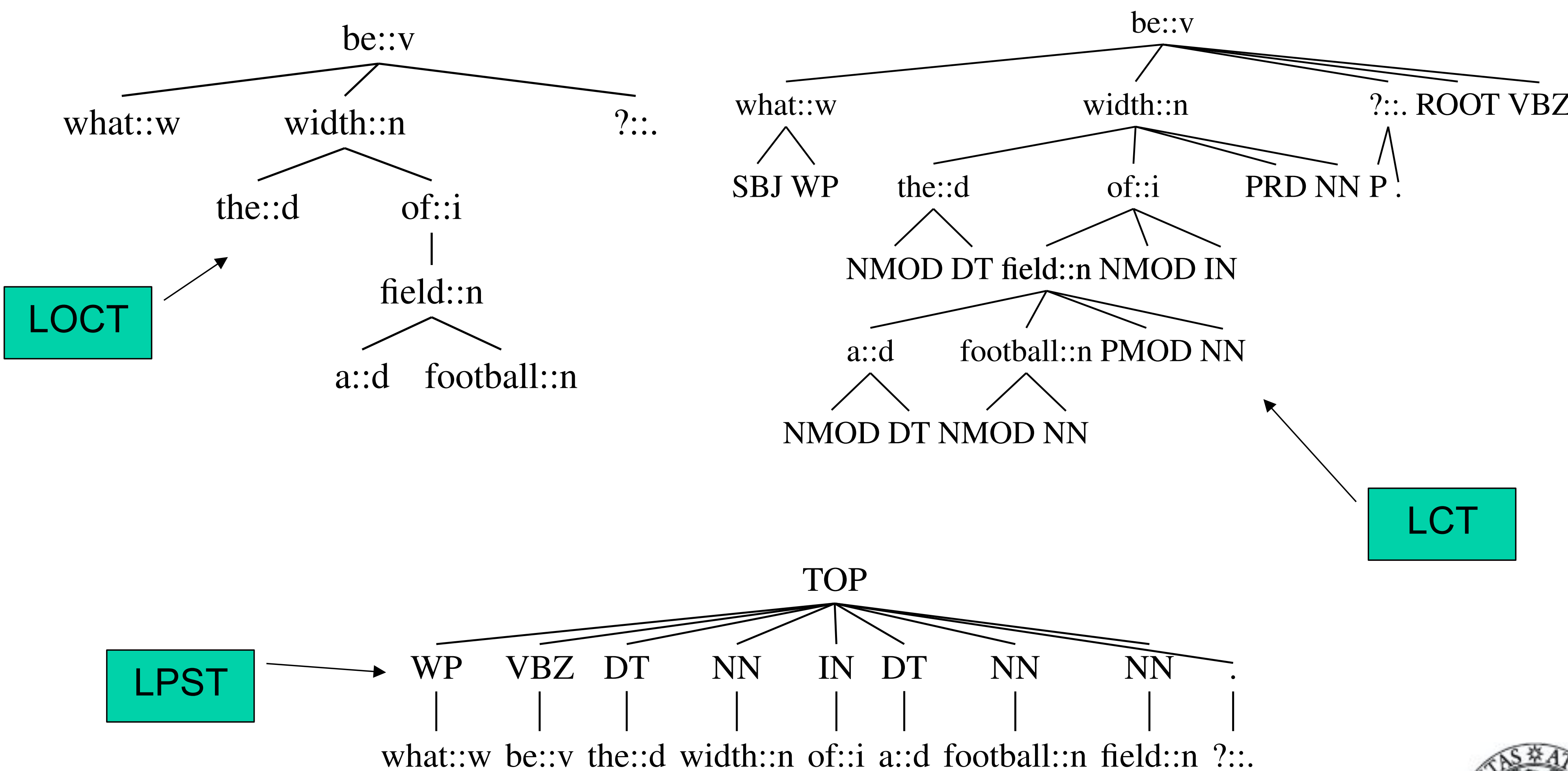
Lin:

$$sim_{LIN}(c_1, c_2) = \frac{2\, \log\, P(lso(c_1, c_2))}{\log\, P(c_1) + \log P\,(c_2)}$$

# Question Classification with SSTK
## [Blohedorn&Moschitti, CIKM2007]

| | **Accuracy** | | | | |
|---|---|---|---|---|---|
| **$\lambda$ parameter** | **0.4** | **0.05** | **0.01** | **0.005** | **0.001** |
| **linear (bow)** | 0.905 | | | | |
| **string matching** | 0.890 | 0.910 | **0.914** | **0.914** | 0.912 |
| **full** | 0.904 | **0.924** | 0.918 | 0.922 | 0.920 |
| **full-ic** | 0.908 | **0.922** | 0.916 | 0.918 | 0.918 |
| **path-1** | 0.906 | **0.918** | 0.912 | **0.918** | 0.916 |
| **path-2** | 0.896 | 0.914 | 0.914 | **0.916** | **0.916** |
| **lin** | 0.908 | **0.924** | 0.918 | 0.922 | 0.922 |
| **wup** | 0.908 | **0.926** | 0.918 | 0.922 | 0.922 |

# Same Task with PTK, SPTK and Dependency Trees



# Results
## [Croce, Moschitti, Basili, EMNLP 2011]

| | STK | PTK | SPTK(LSA) |
|---|---|---|---|
| CT | 91.20% | 90.80% | 91.00% |
| LOCT | - | 89.20% | 93.20% |
| LCT | - | 90.80% | **94.80%** |
| LPST | - | 89.40% | 89.60% |
| BOW | | 88.80% | |


JEOPARDY!

# Classification in Definition vs not Definition in Jeopardy

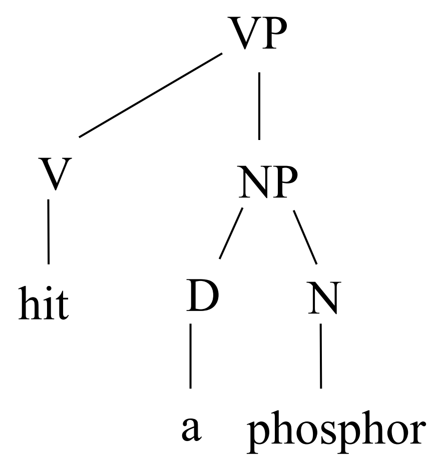
- **Definition:** *Usually, to do this is to lose a game without playing it*

  (solution: *forfeit*)
- **Non Definition:** *When hit by electrons, a phosphor gives off electromagnetic energy in this form*
- Complex linguistic problem: let us learning it with syntactic similarity from training examples

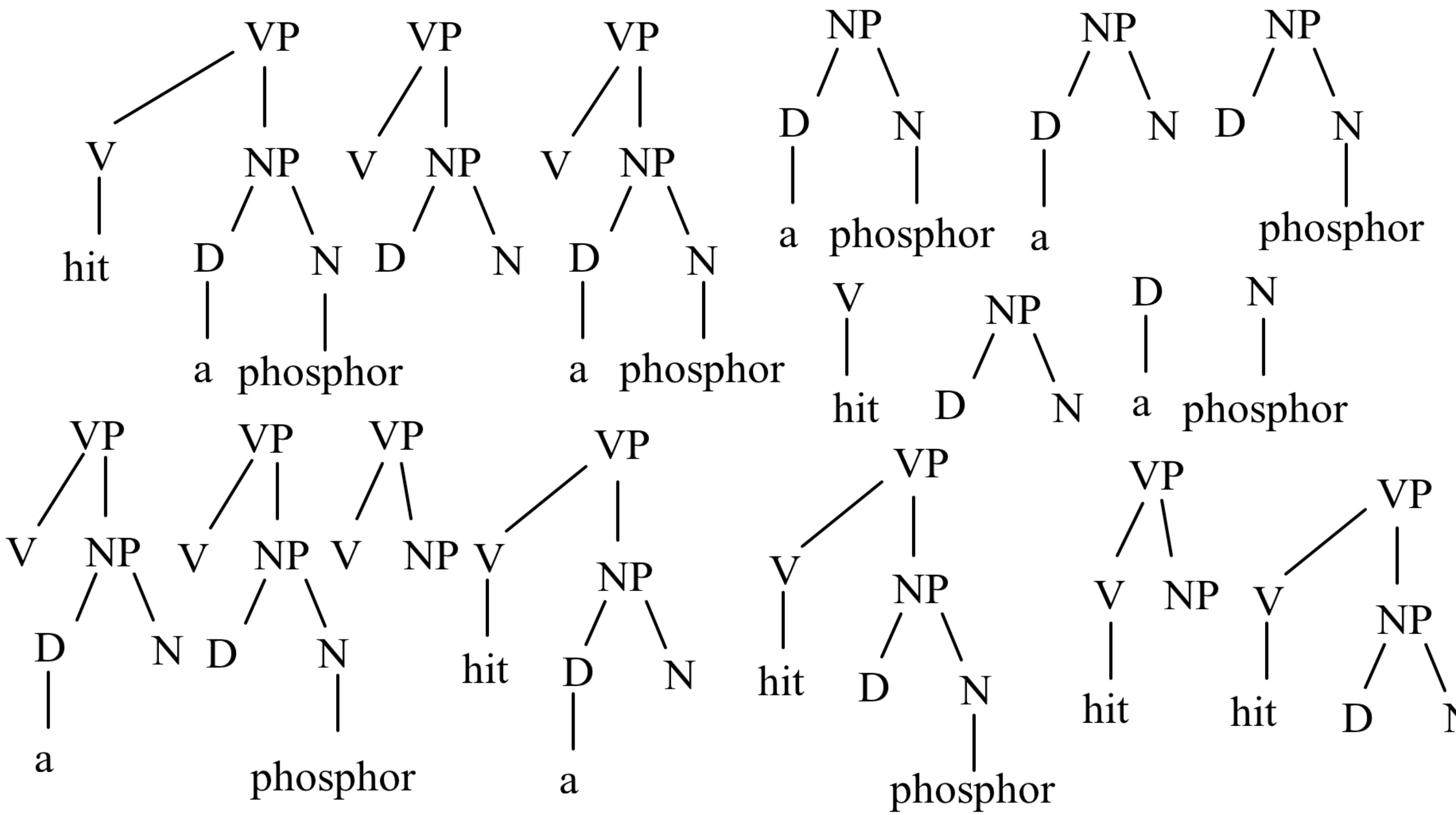
# Automatic Learning of a Question Classifier

- Similarity between definitions vs similarity between non definition
- Instead of using features-based similarity we used kernels
- Combining several linguistic structures with several kernels for representing a question **q**:
  - $K_1(\langle q_1,q_2\rangle)+K_2(\langle q_1,q_2\rangle)+\ldots+K_n(\langle q_1,q_2\rangle)$
- Tree kernels measures similarity between trees

# Syntactic Tree Kernel (STK)
# (Collins and Duffy 2002)

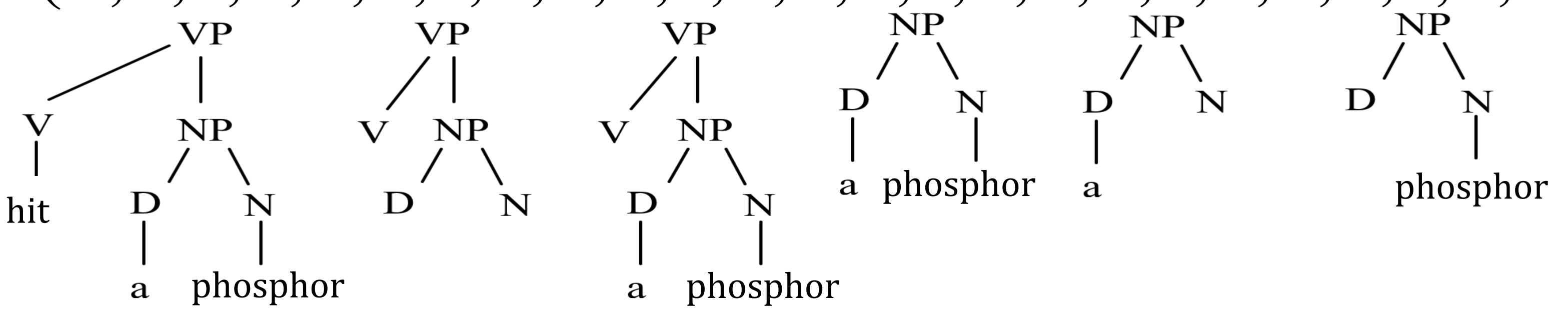


# Syntactic Tree Kernel (STK) (Collins and Duffy 2002)

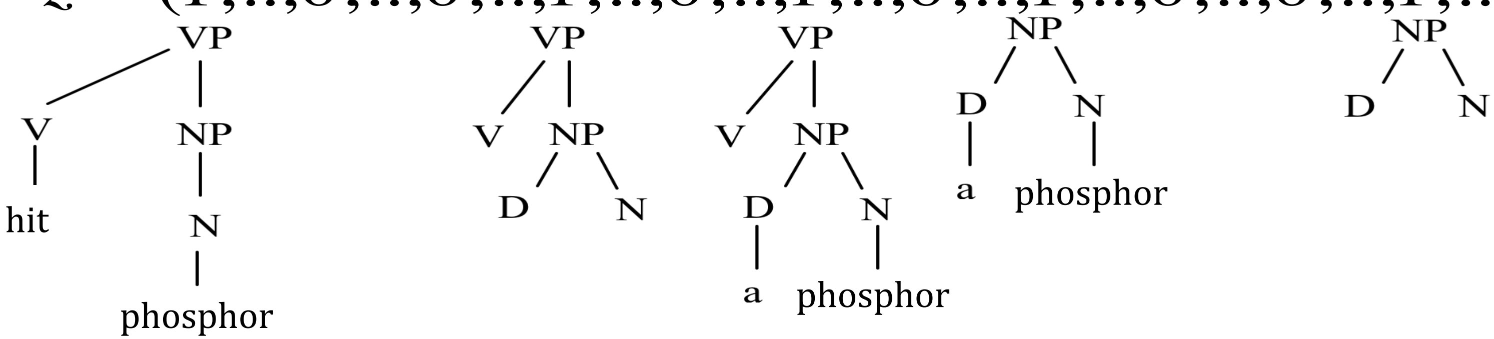


# The resulting explicit kernel space

$$\phi(T_x) = \vec{x} = (0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,1,..,0)$$




$$\phi(T_z) = \vec{z} = (1,..,0,..,0,..,1,..,0,..,1,..,0,..,1,..,0,..,0,..,1,..,0,..,0)$$

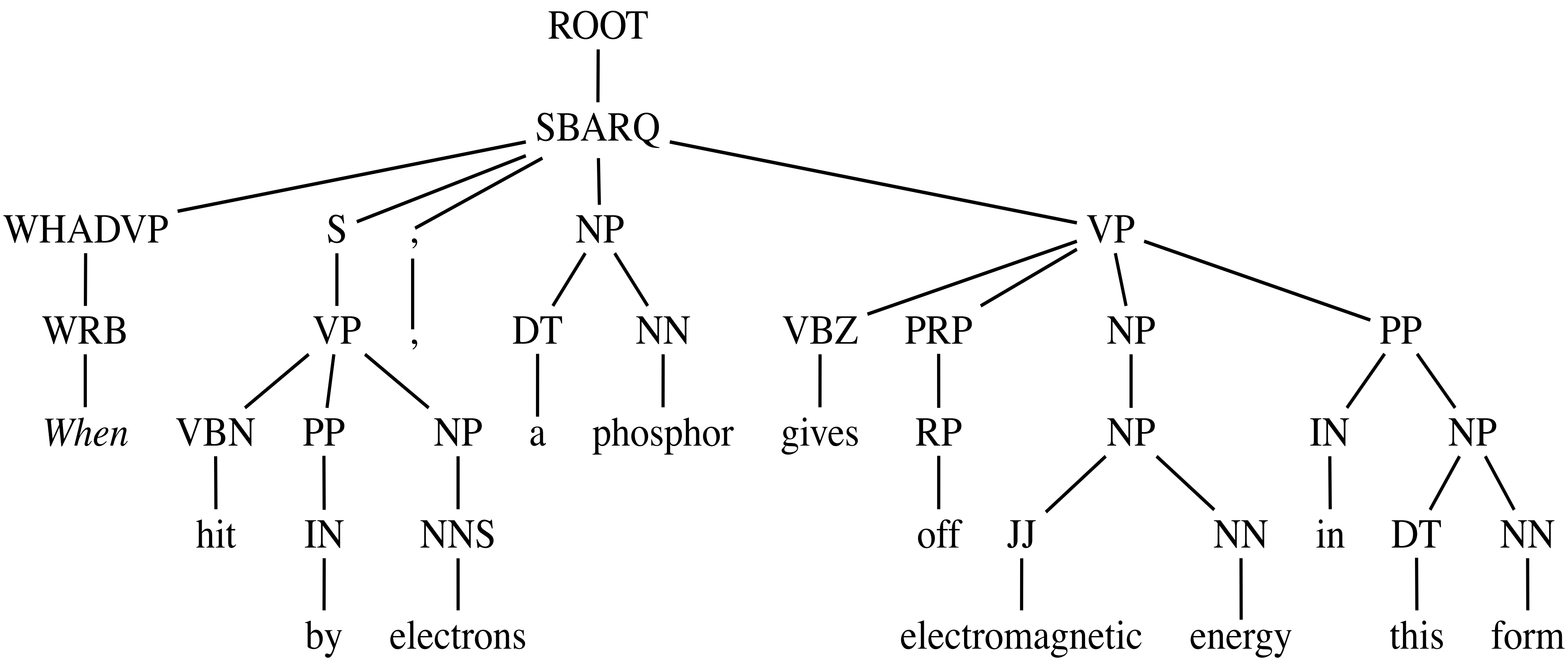



- $\vec{x} \cdot \vec{z}$ counts the number of common substructures

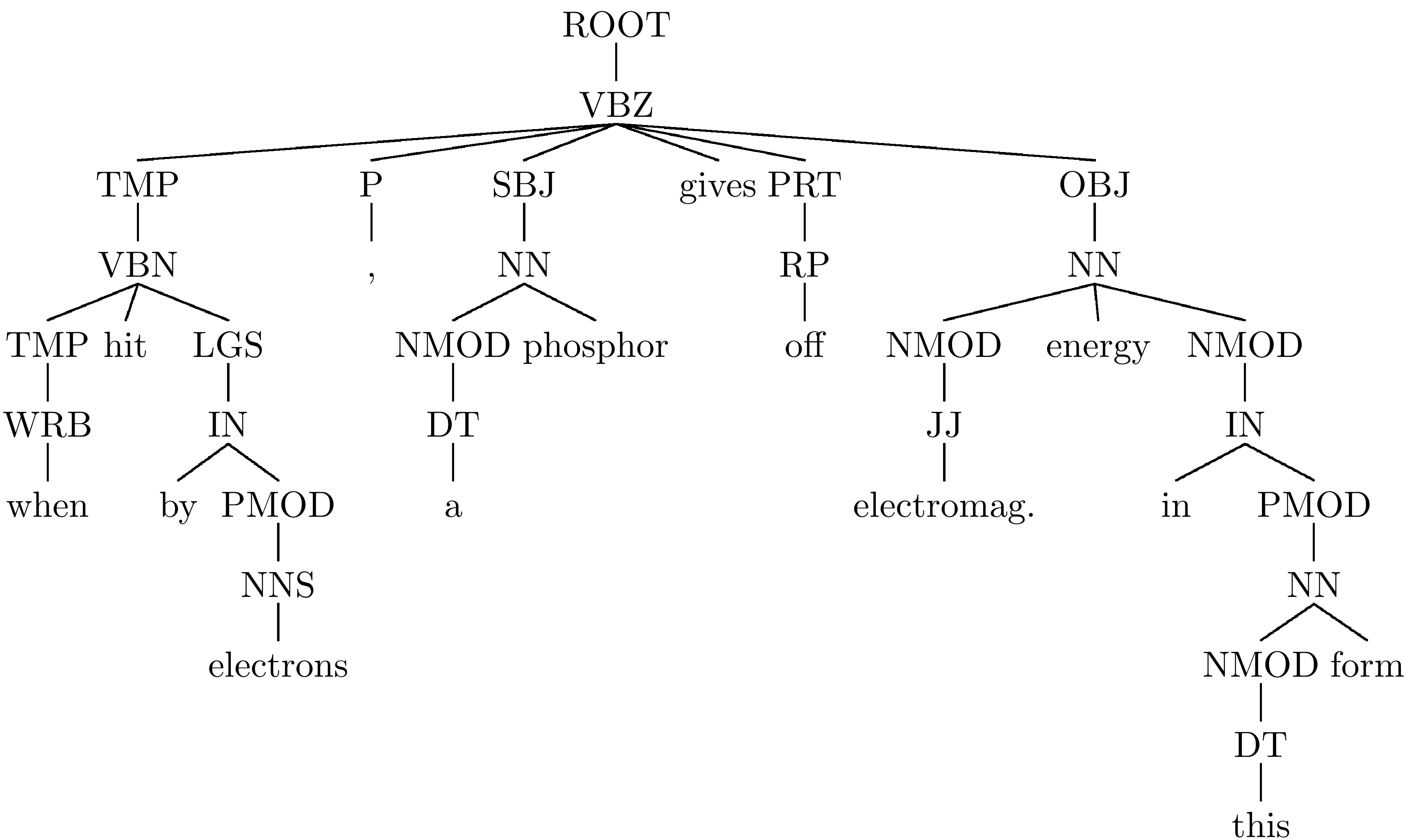
# Experimental setup

- Corpus: a random sample from 33 Jeopardy! Games
- 306 definition and 4,964 non-definition clues
- Tools:
  - SVMLight-TK
  - Charniak's constituency parser
  - Syntactic/Semantic parser by Johansson and Nugues (2008)
- Measures derived with leave-on-out

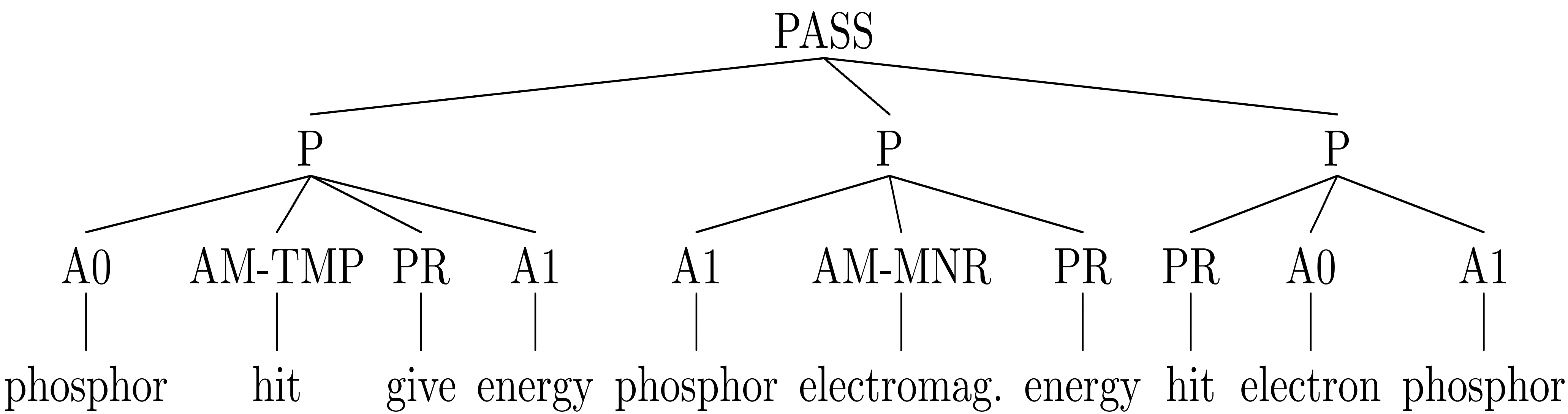
# Constituency Tree (CT)



# Dependency Tree (DT)



# Predicate Argument Structure Set (**PASS**)



# Sequence Kernels

**WSK**: [when][hit][by][electrons][,][a][phosphor][gives][off][electromagnetic][energy][in][this][form]

**PSK**: [wrb][vbn][in][nns][,][dt][nn][vbz][rp][jj][nn][in][dt][nn]

**CSK**: [general][science]
(category sequence kernel)

# Individual models

| Kernel Space | Prec. | Rec. | F1 |
|---|---|---|---|
| RBC | 28.27 | 70.59 | **40.38** |
| BOW | 47.67 | 46.73 | 47.20 |
| WSK | 47.11 | 50.65 | 48.82 |
| STK-CT | 50.51 | 32.35 | 39.44 |
| PTK-CT | 47.84 | 57.84 | **52.37** |
| PTK-DT | 44.81 | 57.84 | 50.50 |
| PASS | 33.50 | 21.90 | 26.49 |
| PSK | 39.88 | 45.10 | 42.33 |
| CSK | 39.07 | 77.12 | **51.86** |

# Model Combinations

| Kernel Space | Prec. | Rec. | F1 |
|---|---|---|---|
| WSK+CSK | 70.00 | 57.19 | 62.95 |
| PTK-CT+CSK | 69.43 | 60.13 | 64.45 |
| PTK-CT+WSK+CSK | 68.59 | 62.09 | **65.18** |
| BOW+CSK+RBC | 60.65 | 73.53 | 66.47 |
| PTK-CT+WSK+CSK+RBC | 67.66 | 66.99 | **67.32** |
| PTK-CT+PASS+CSK+RBC | 62.46 | 71.24 | 66.56 |
| WSK+CSK+RBC | 69.26 | 66.99 | **68.11** |
| ALL | 61.42 | 67.65 | 64.38 |

**66.7% of relative improvement on RBC**

# Impact of QC in Watson

- Specific evaluation on definition questions
  - 1,000 unseen games (60,000 questions)
  - Two test sets of 1,606 and 1,875 questions derived with:
    - Statistical model (StatDef)
    - RBC (RuleDef)
  - Direct comparison only with NoDef
- All questions evaluation
  - Selected 66 unseen Jeopardy! games
  - 3,546 questions

# Watson's Accuracy, Precision and Earnings

- Comparison between use or not QC
- Different set of questions

| | NoDef | StatDef | NoDef | RuleDef |
|---|---|---|---|---|
| # Questions | 1606 | 1606 | 1875 | 1875 |
| Accuracy | 63.76% | 65.57% | 56.64% | 57.51% |
| P@70 | 82.22% | 84.53% | 72.73% | 74.87% |

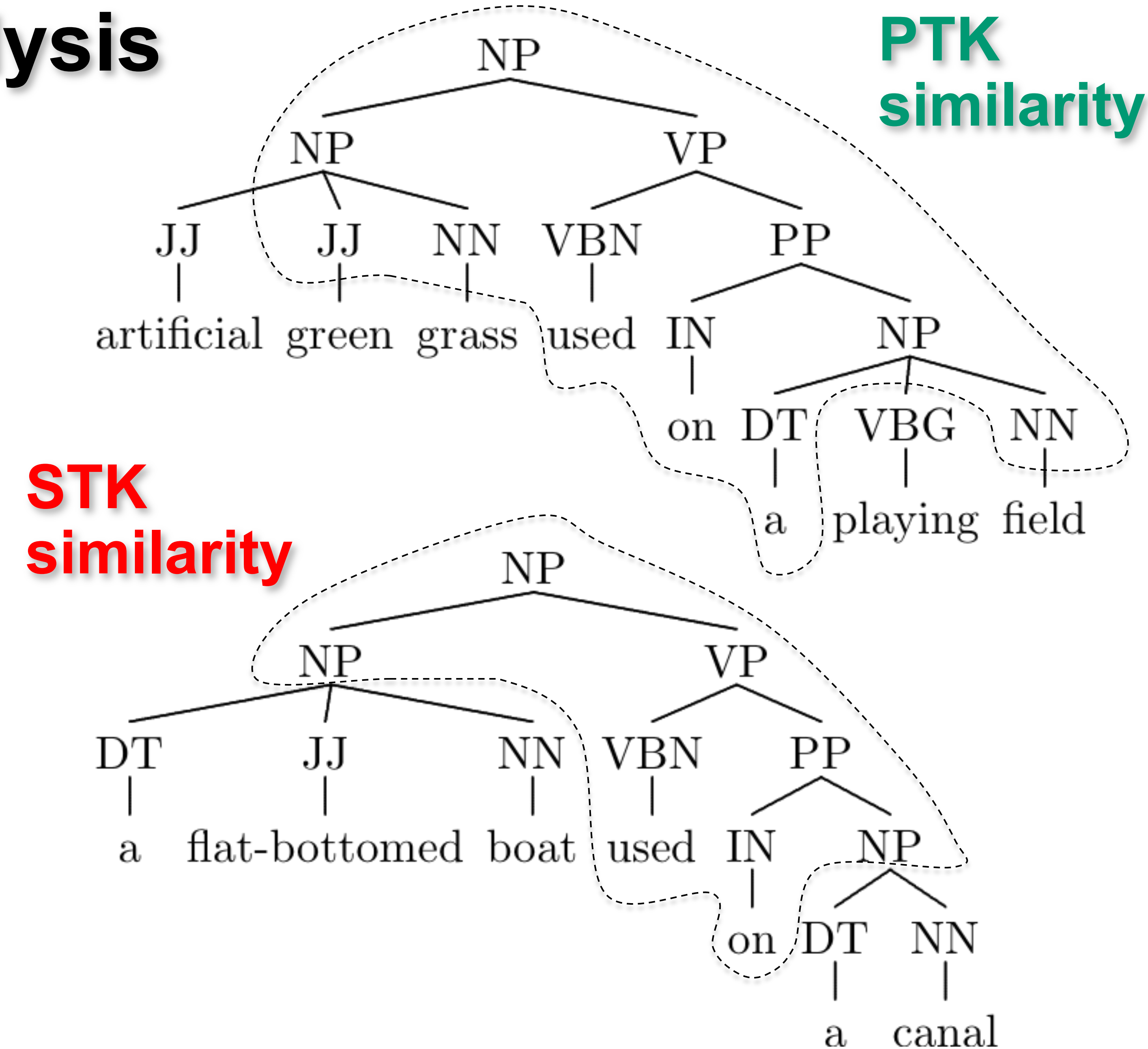
| | # Def Q's | Accuracy | P@70 | Earnings |
|---|---|---|---|---|
| NoDef | 0 | 69.71% | 86.79% | $24,818 |
| RuleDef | 480 | 69.23% | 86.31% | $24,397 |
| StatDef | 131 | 69.85% | 87.19% | $25,109 |

# Error Analysis

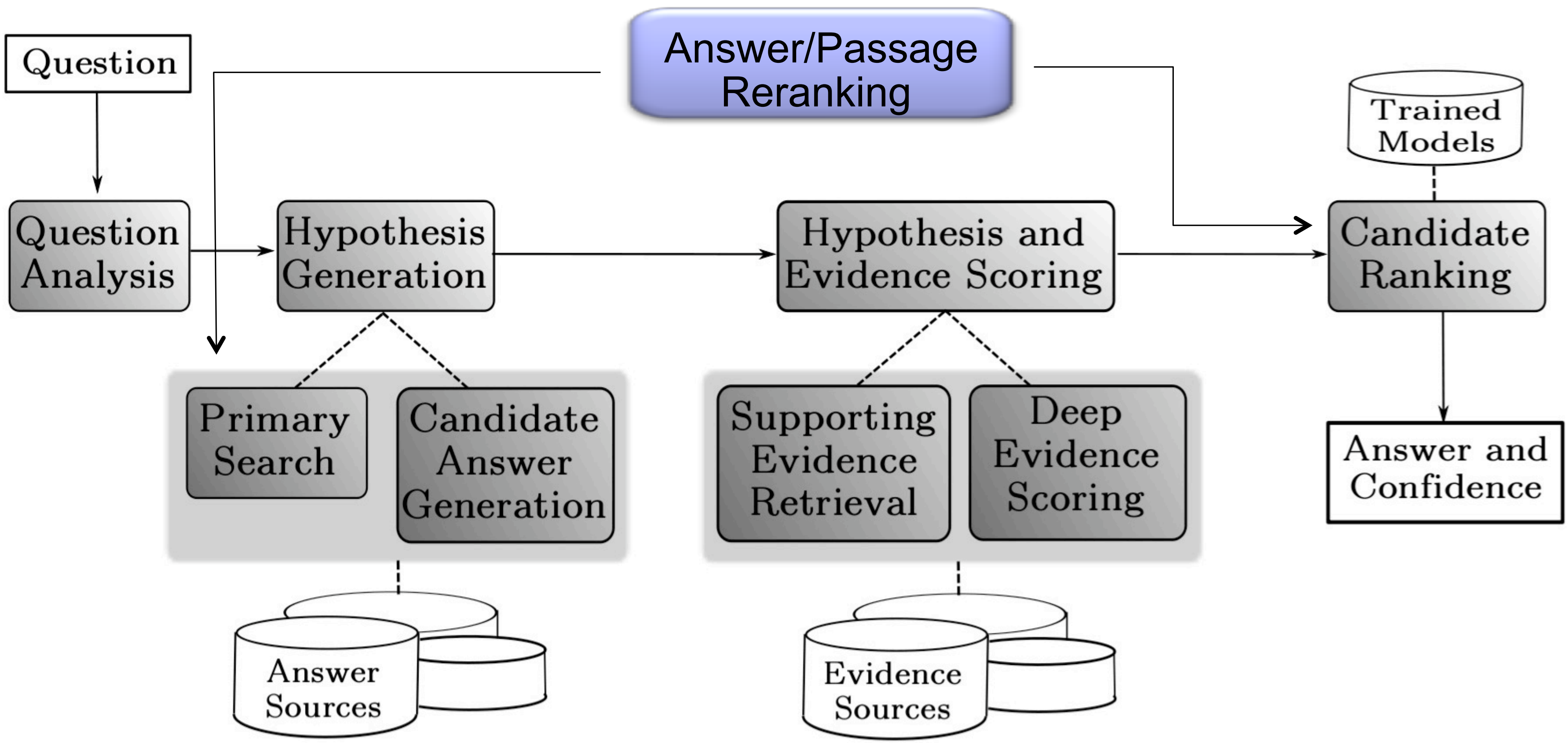
## Test Example

- PTK ok
- STK not ok

## Training Example



# Answer/Passage Reranking

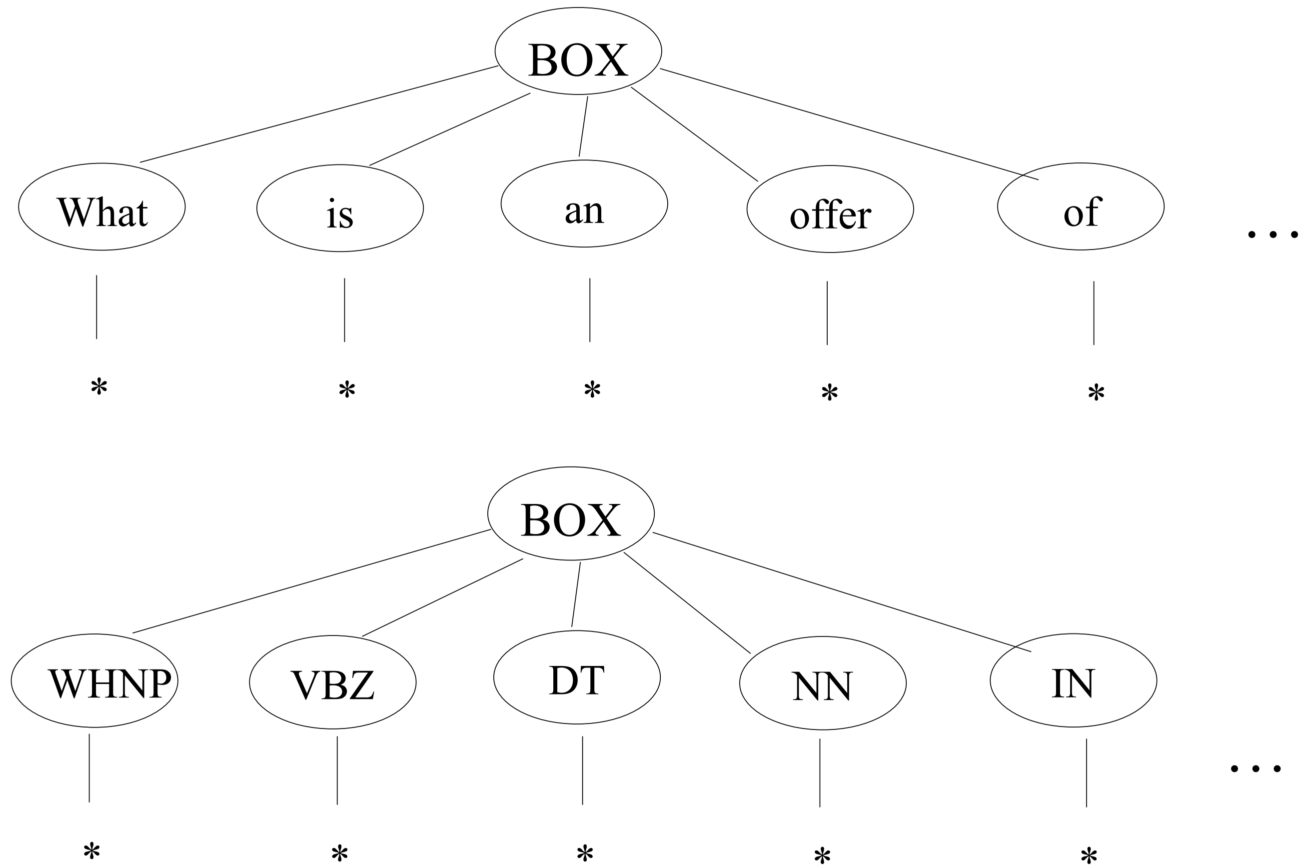


# TASK: Question/Answer Classification
## [Moschitti, CIKM 2008]

- The classifier detects if a pair (question and answer) is correct or not
- A representation for the pair is needed
- The classifier can be used to re-rank the output of a basic QA system

# Bags of words (BOW) and POS-tags (POS)

- To save time, apply tree kernels to these trees:

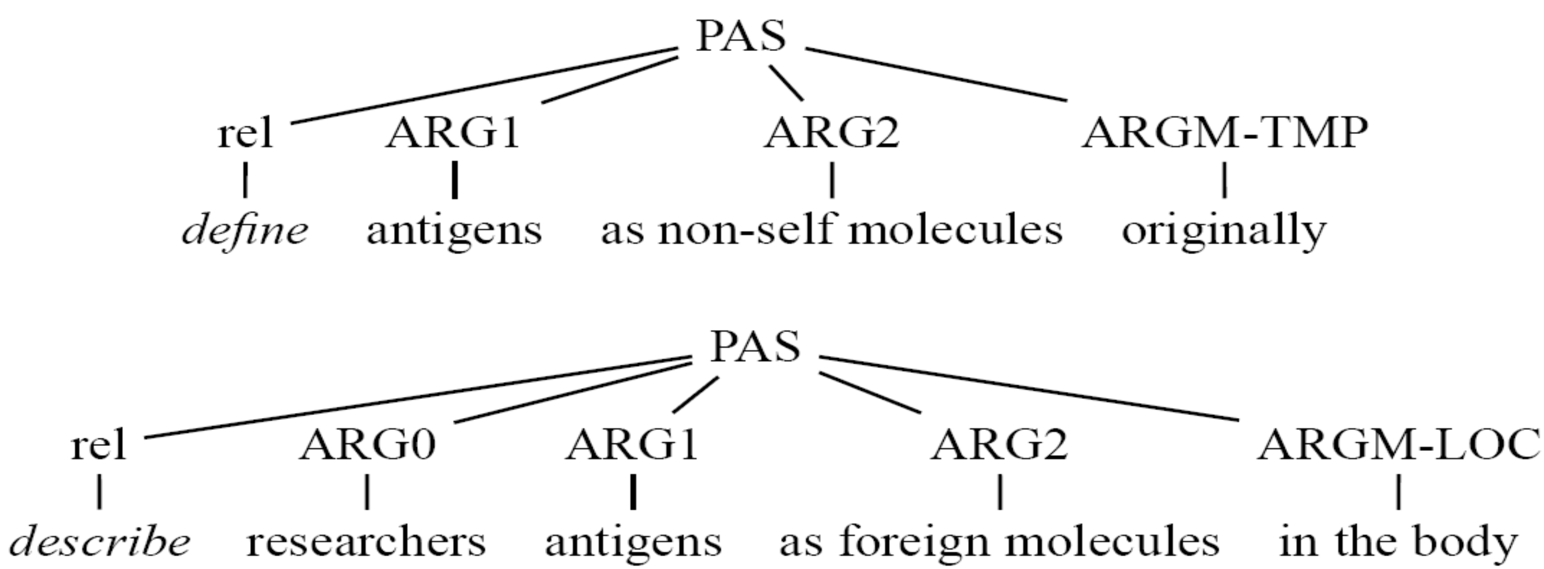


# Word and POS Sequences

- What is an offer of…? (word sequence, **WSK**)
  - → `What_is_offer`
  - → `What_is`
- WHNP VBZ DT NN IN…(POS sequence, **POSSK**)
  - → `WHNP_VBZ_NN`
  - → `WHNP_NN_IN`

# Predicate Argument Structures for describing answers ($\text{PAS}_{\text{PTK}}$)

- [**ARG1** Antigens] were [**AM–TMP** originally] [**rel** defined] [**ARG2** as non-self molecules].
- [**ARG0** Researchers] [**rel** describe] [**ARG1** antigens][**ARG2** as foreign molecules] [**ARGM–LOC** in the body]



# Dataset 2: TREC data

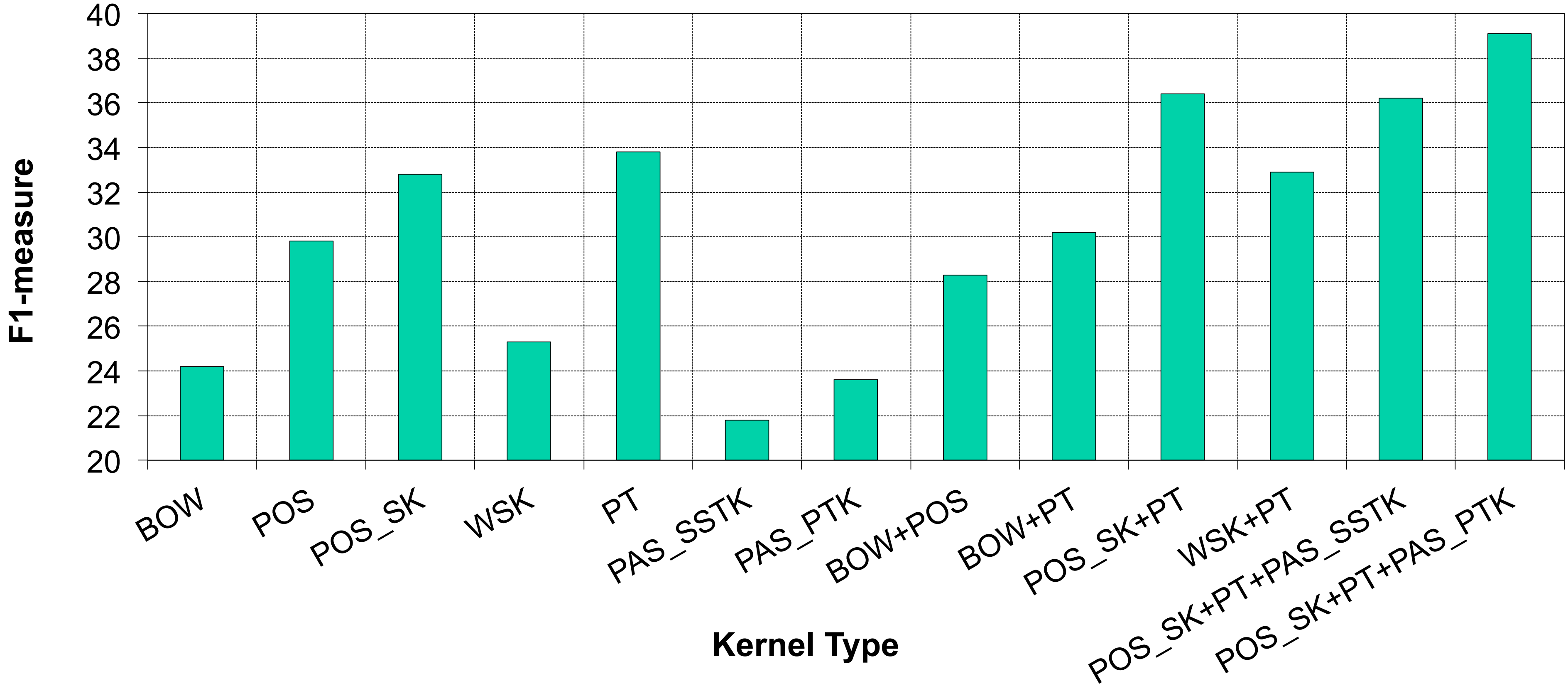
- 138 TREC 2001 test questions labeled as “description”
- 2,256 sentences, extracted from the best ranked paragraphs (using a basic QA system based on Lucene search engine on TREC dataset)
- 216 of which labeled as correct by one annotator

# Kernels and Combinations

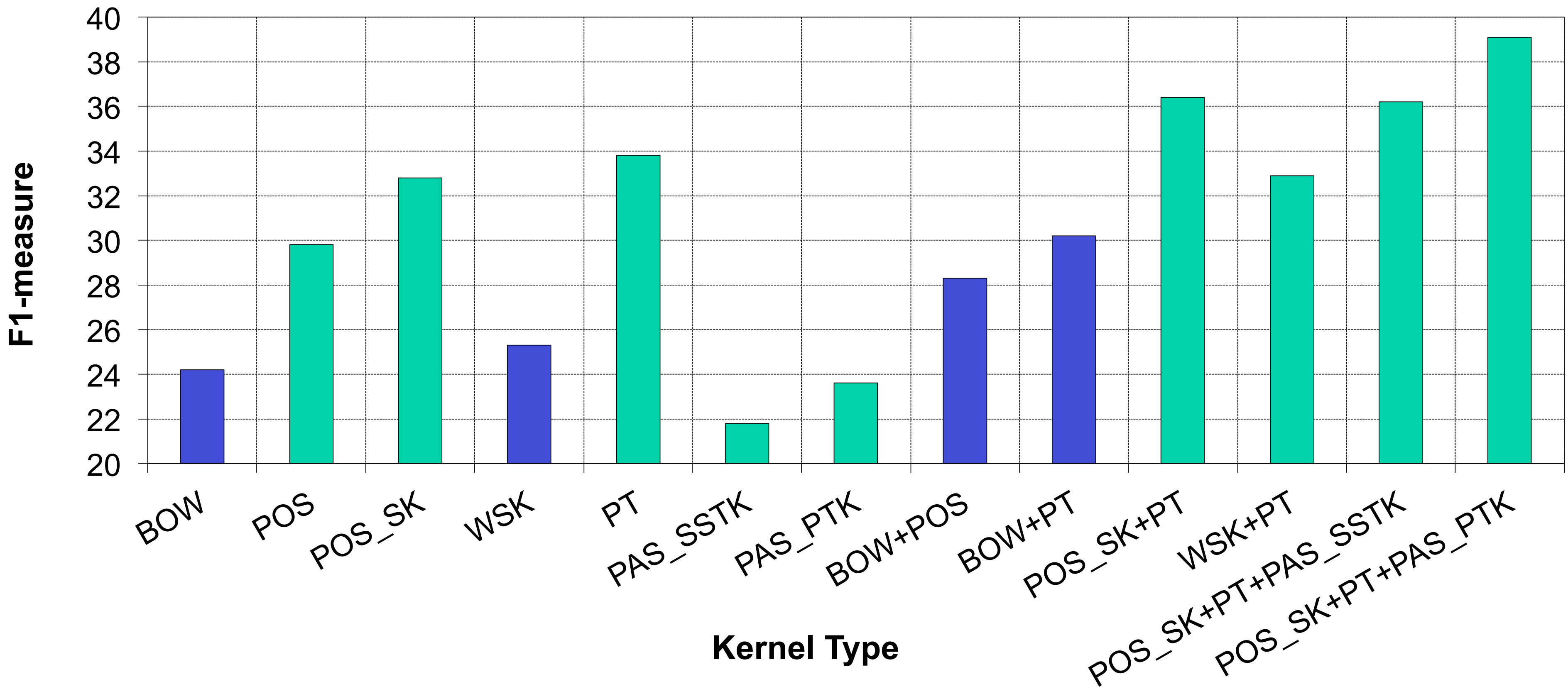
- Exploiting the property: $k(x,z) = k_1(x,z)+k_2(x,z)$
- Given: BOW, POS, WSK, POSSK, PT, $PAS_{PTK}$

$\Rightarrow$ BOW+POS, BOW+PT, PT+POS, …

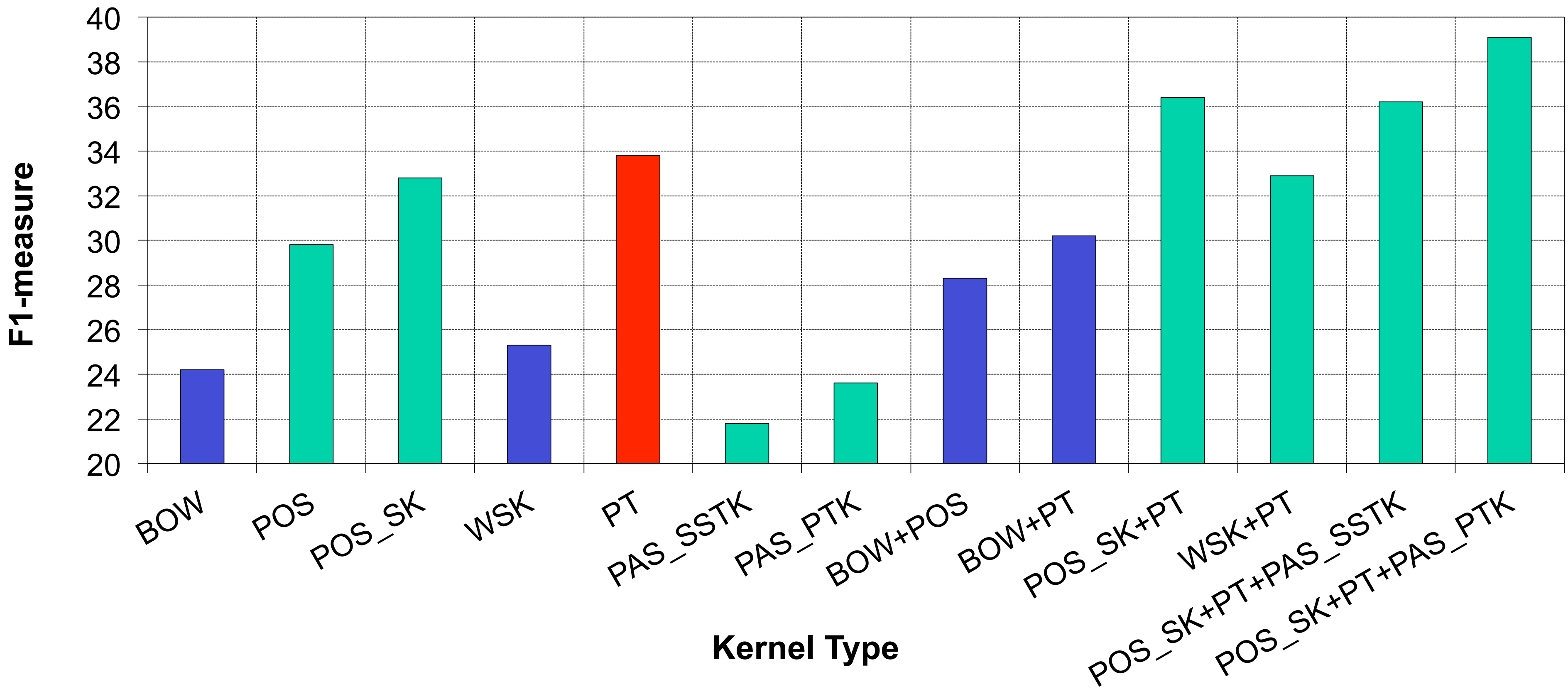
# Results on TREC Data
## (5 folds cross validation)



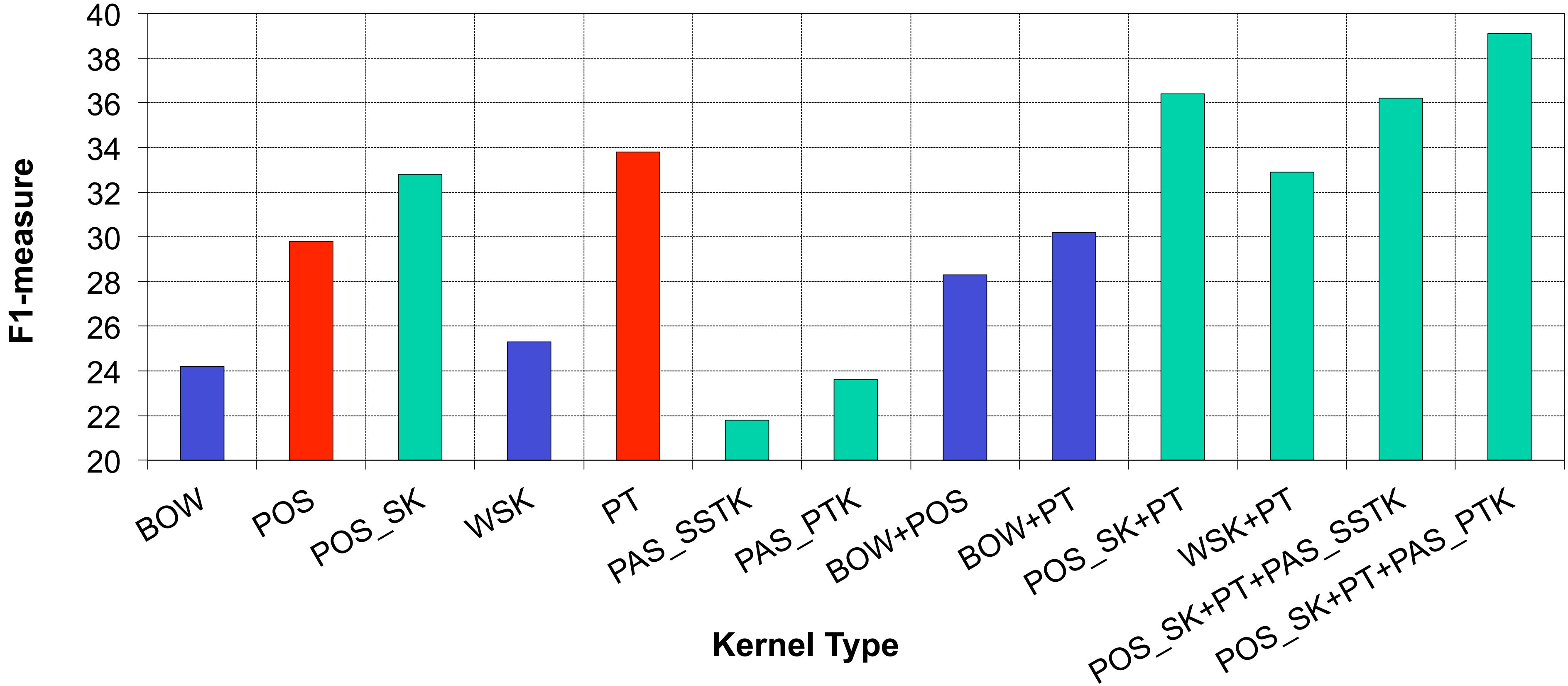
# Results on TREC Data
## (5 folds cross validation)



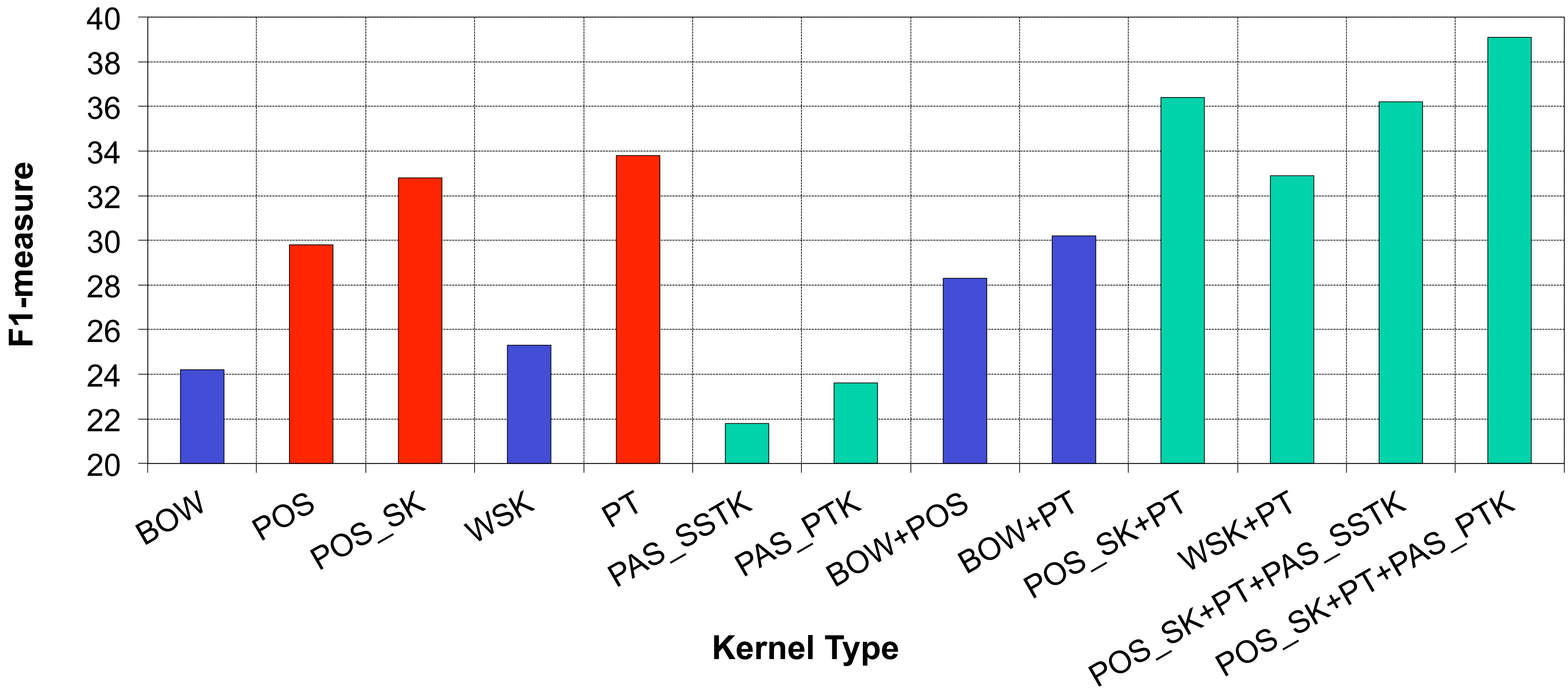
# Results on TREC Data
## (5 folds cross validation)



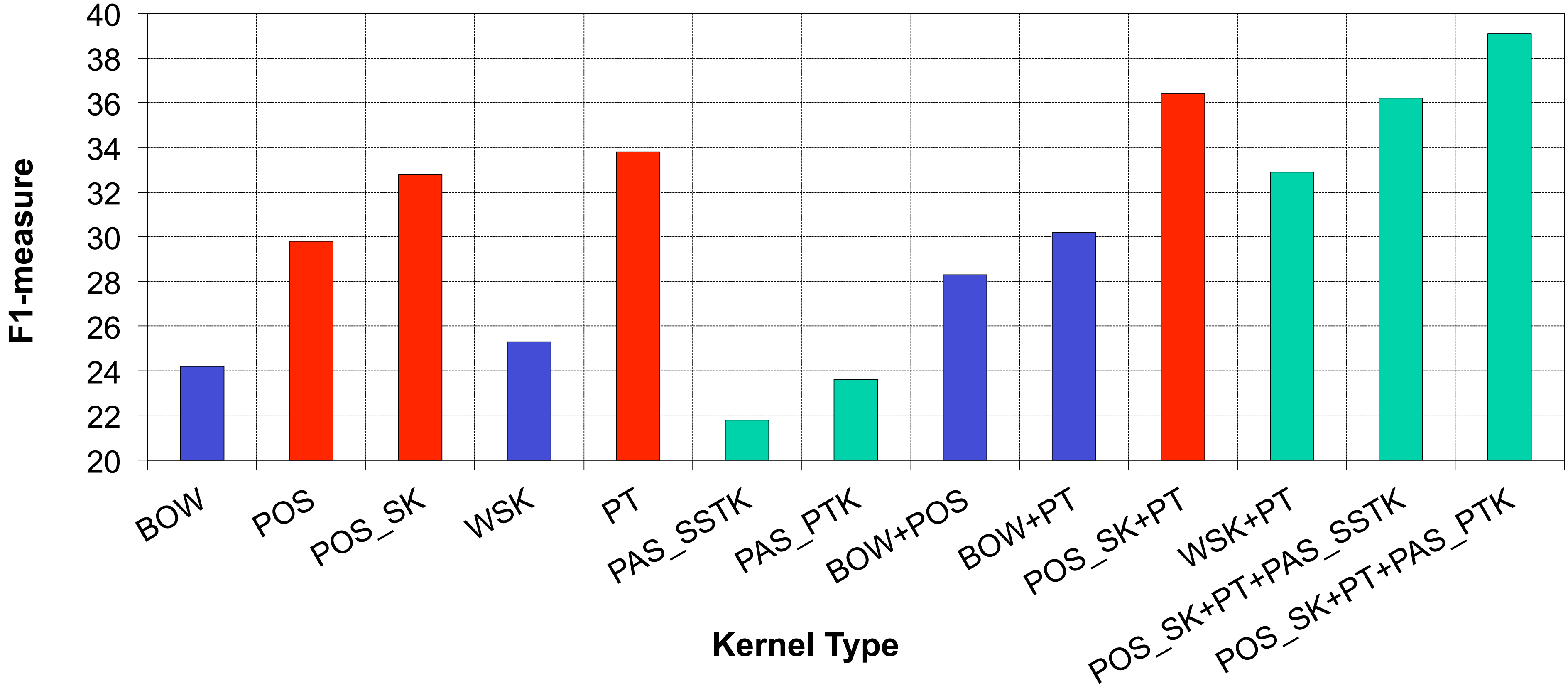
# Results on TREC Data
## (5 folds cross validation)



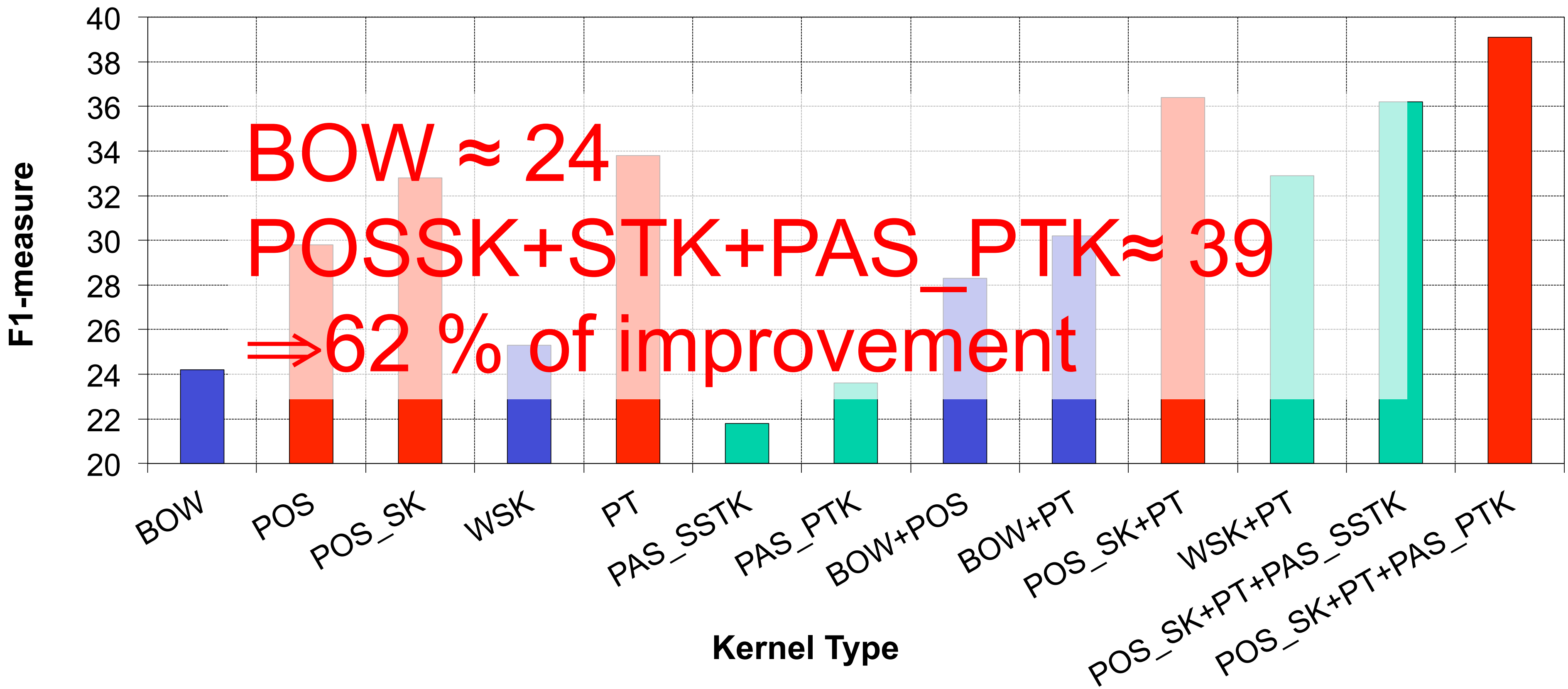
# Results on TREC Data
## (5 folds cross validation)



# Results on TREC Data
## (5 folds cross validation)



# Results on TREC Data
## (5 folds cross validation)

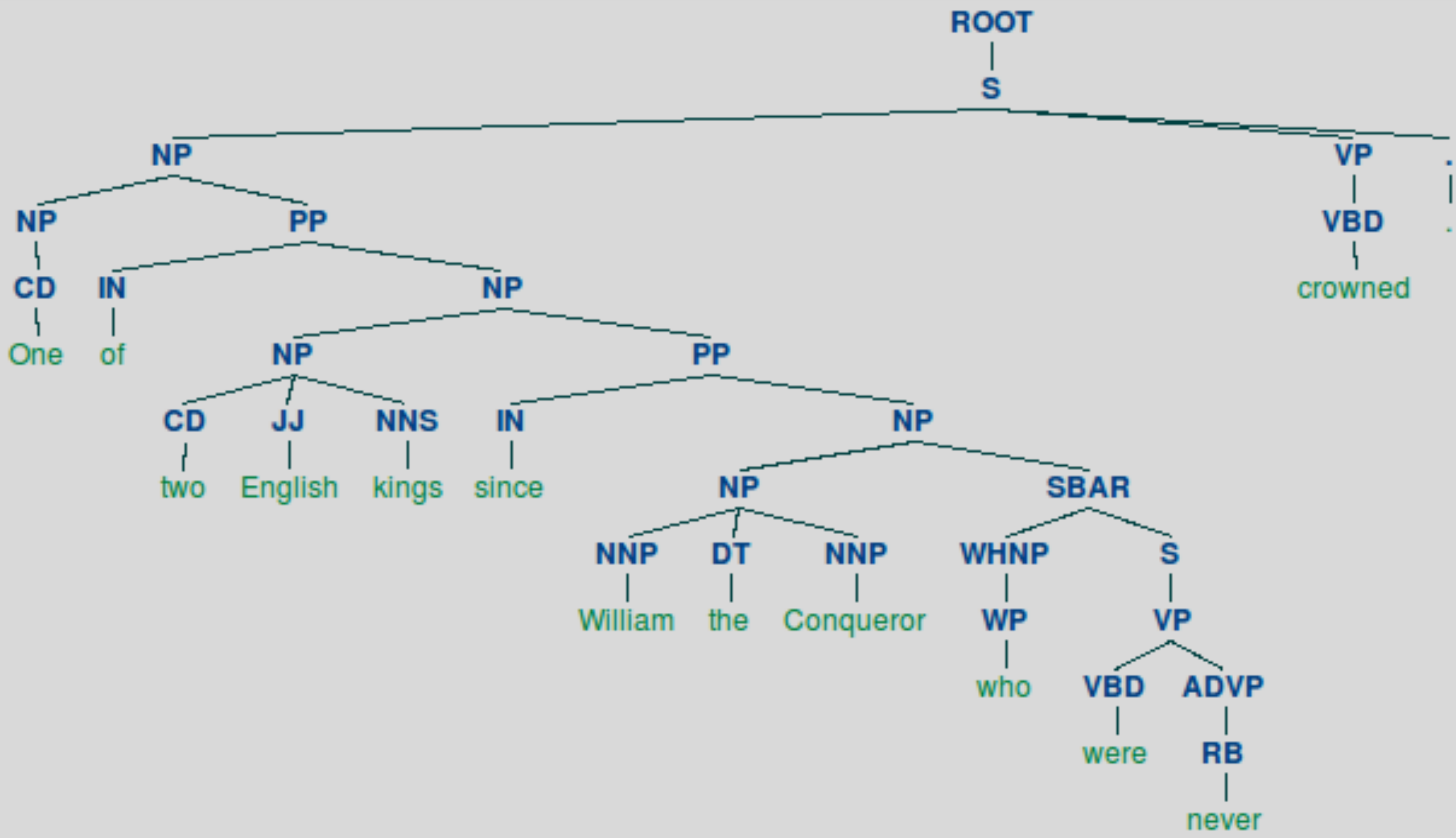


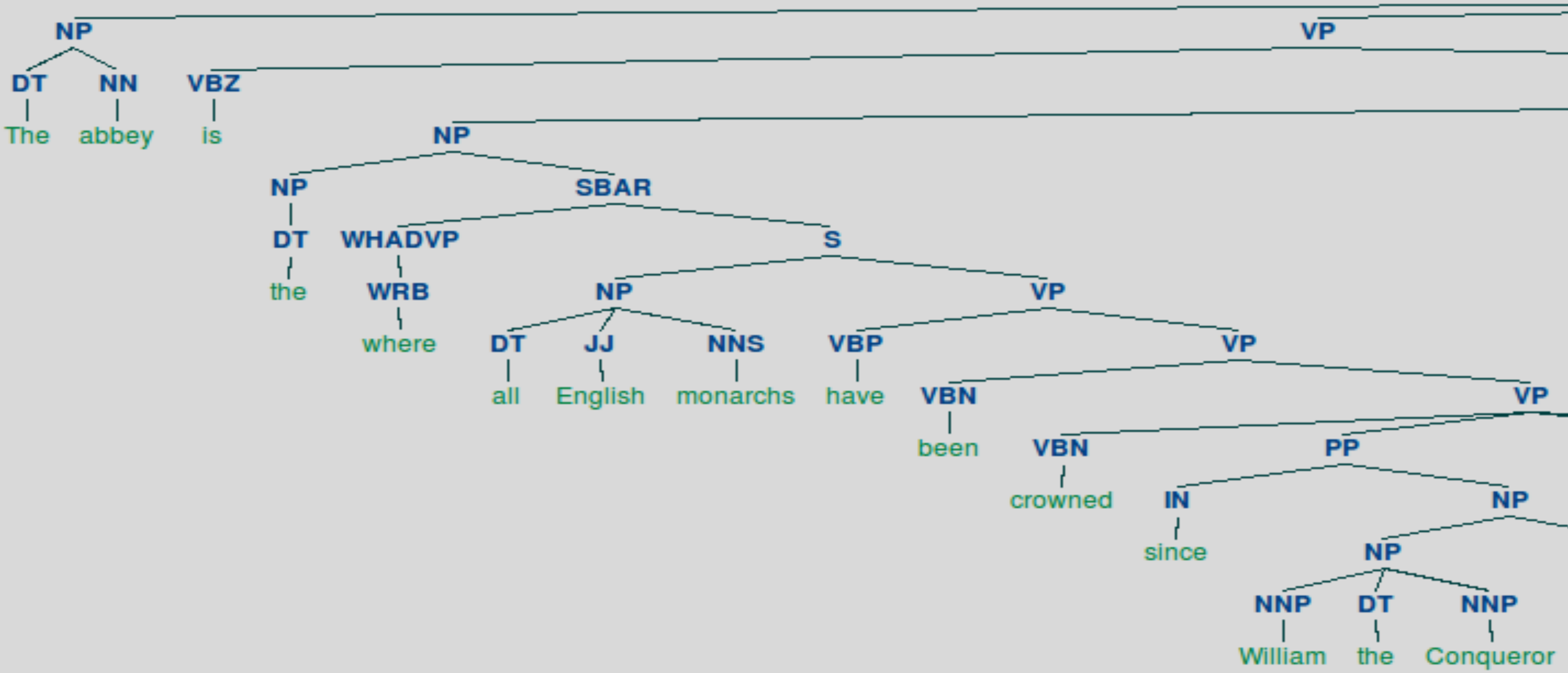

JEOPARDY!

# Our Approach to Answer Selection

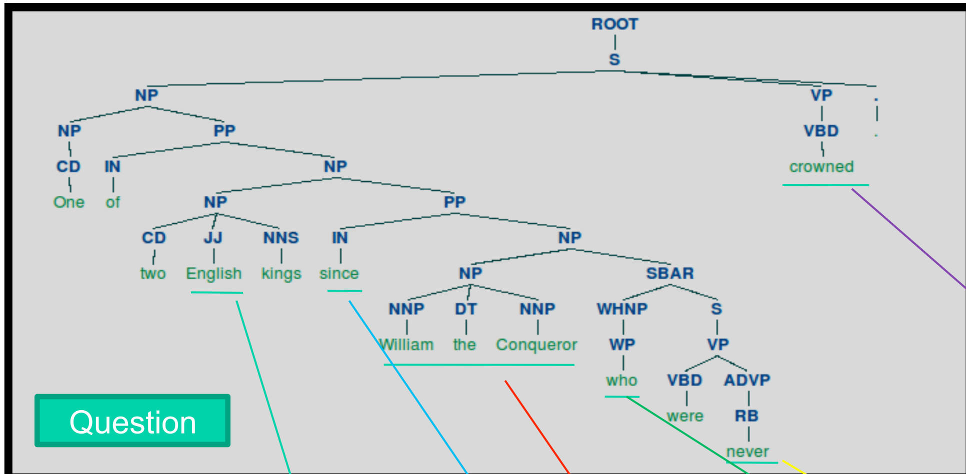
- Learn a classifier of <question,answer> pairs
  - Positive: the answer is correct
  - Negative: otherwise
- Kernel approach
  - Several kernels applied to both questions and answers

# An example of Jeopardy Question



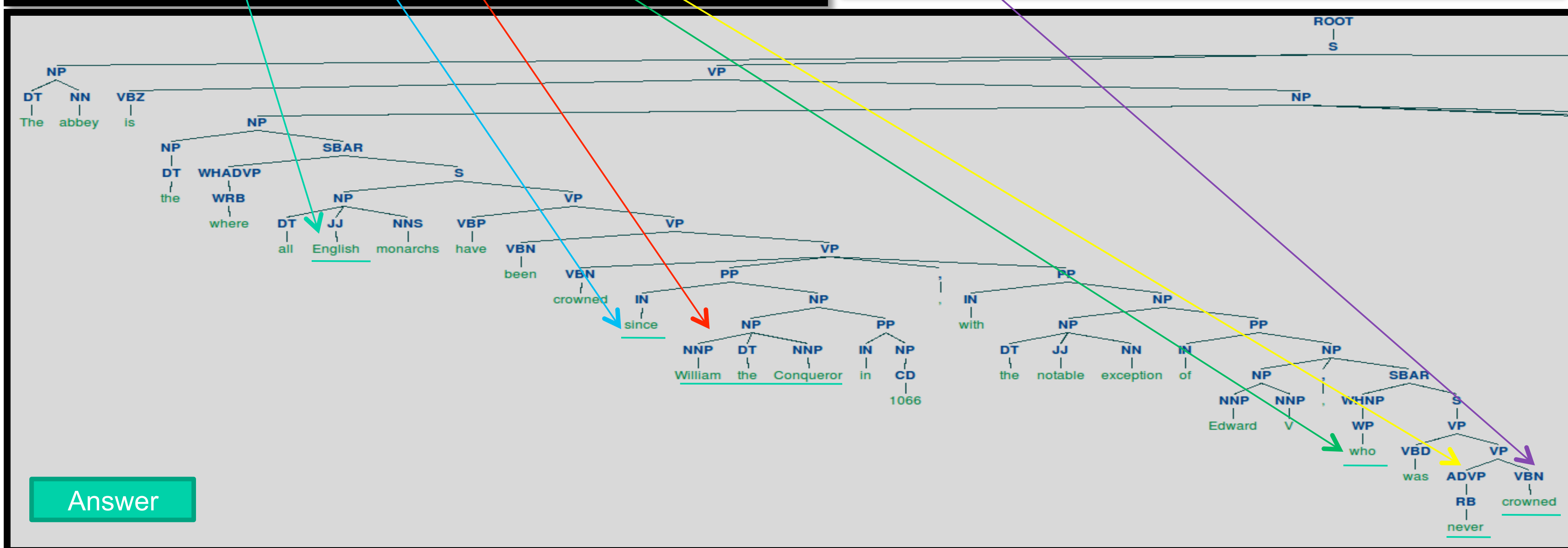

NP
VP
DT
NN
VBZ
The
abbey
is
NP
NP
SBAR
DT
WHADVP
S
the
WRB
NP
VP
where
DT
JJ
NNS
VBP
VP
all
English
monarchs
have
VBN
VP
been
VBN
PP
crowned
IN
NP
since
NP
NNP
DT
NNP
William
the
Conqueror

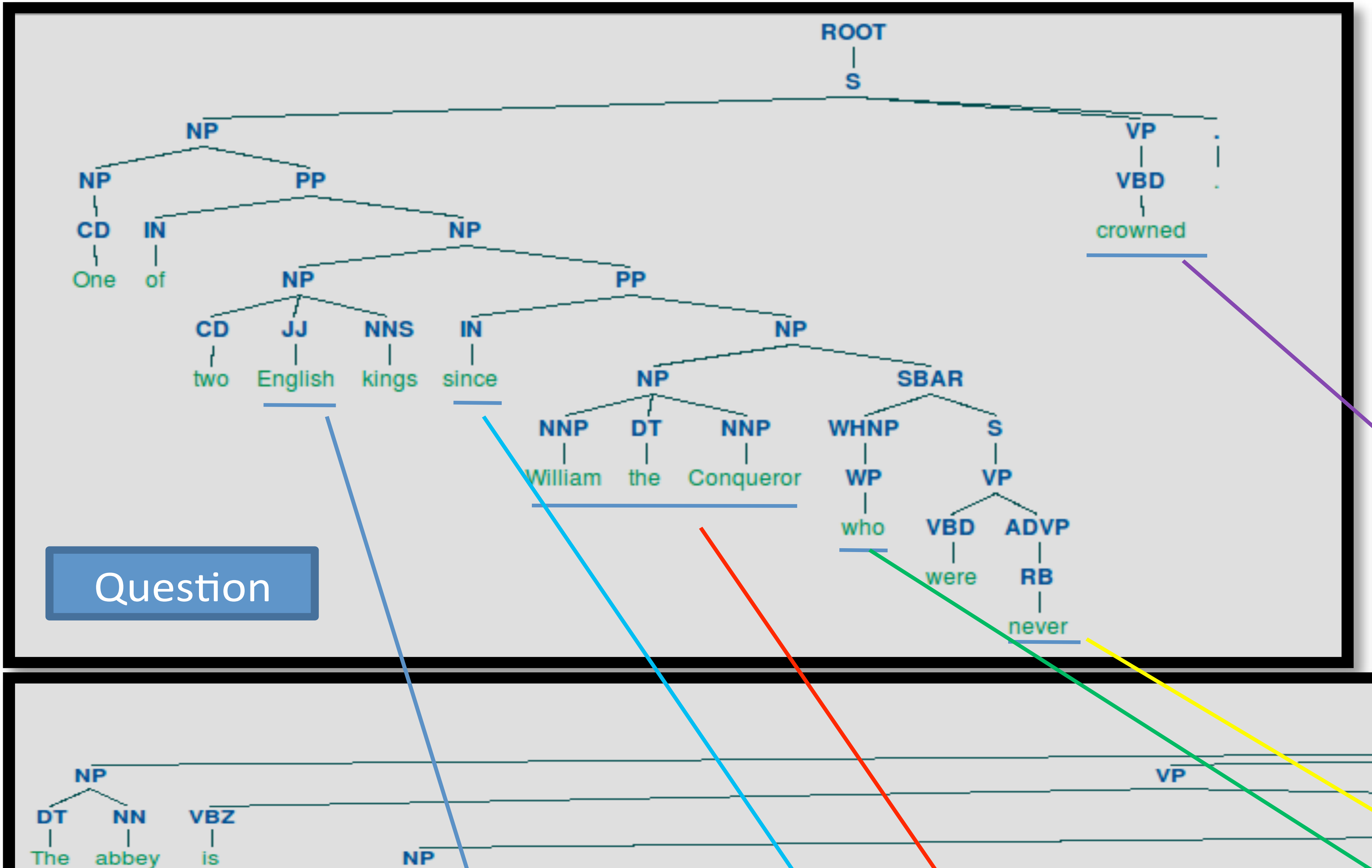
# Baseline Model



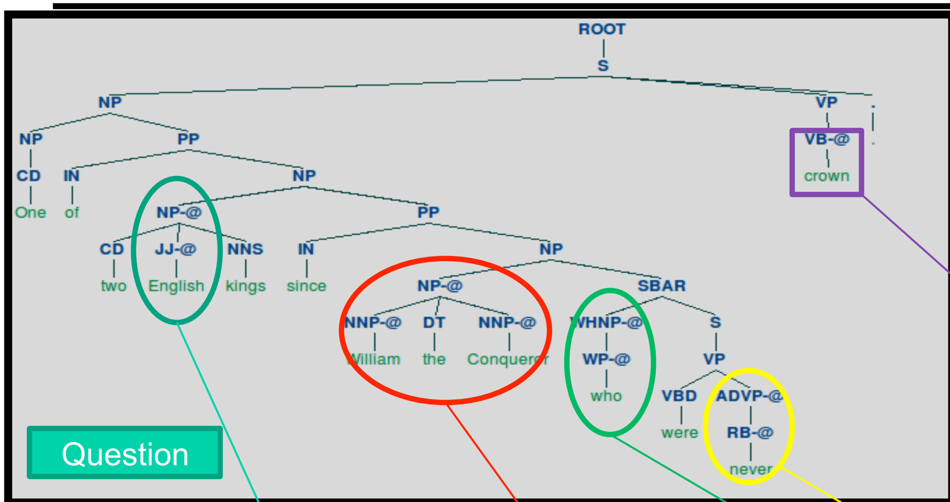

Methodology:

1-Applying PTK without any extra annotation and evaluate the model as baseline.




ROOT
S
NP
VP
.
NP
PP
VBD
CD
IN
NP
crowned
One
of
NP
PP
CD
JJ
NNS
IN
NP
two
English
kings
since
NP
SBAR
NNP
DT
NNP
WHNP
S
William
the
Conqueror
WP
VP
who
VBD
ADVP
were
RB
never
Question
NP
VP
DT
NN
VBZ
The
abbey
is
NP

# Best Model

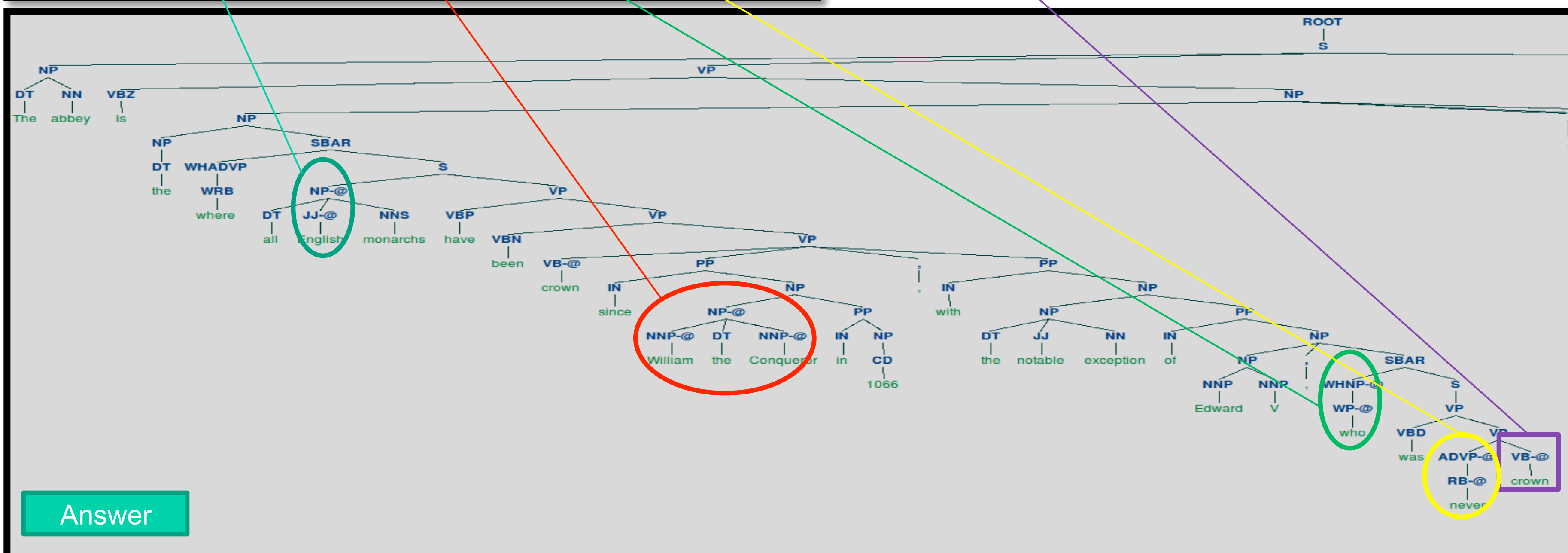


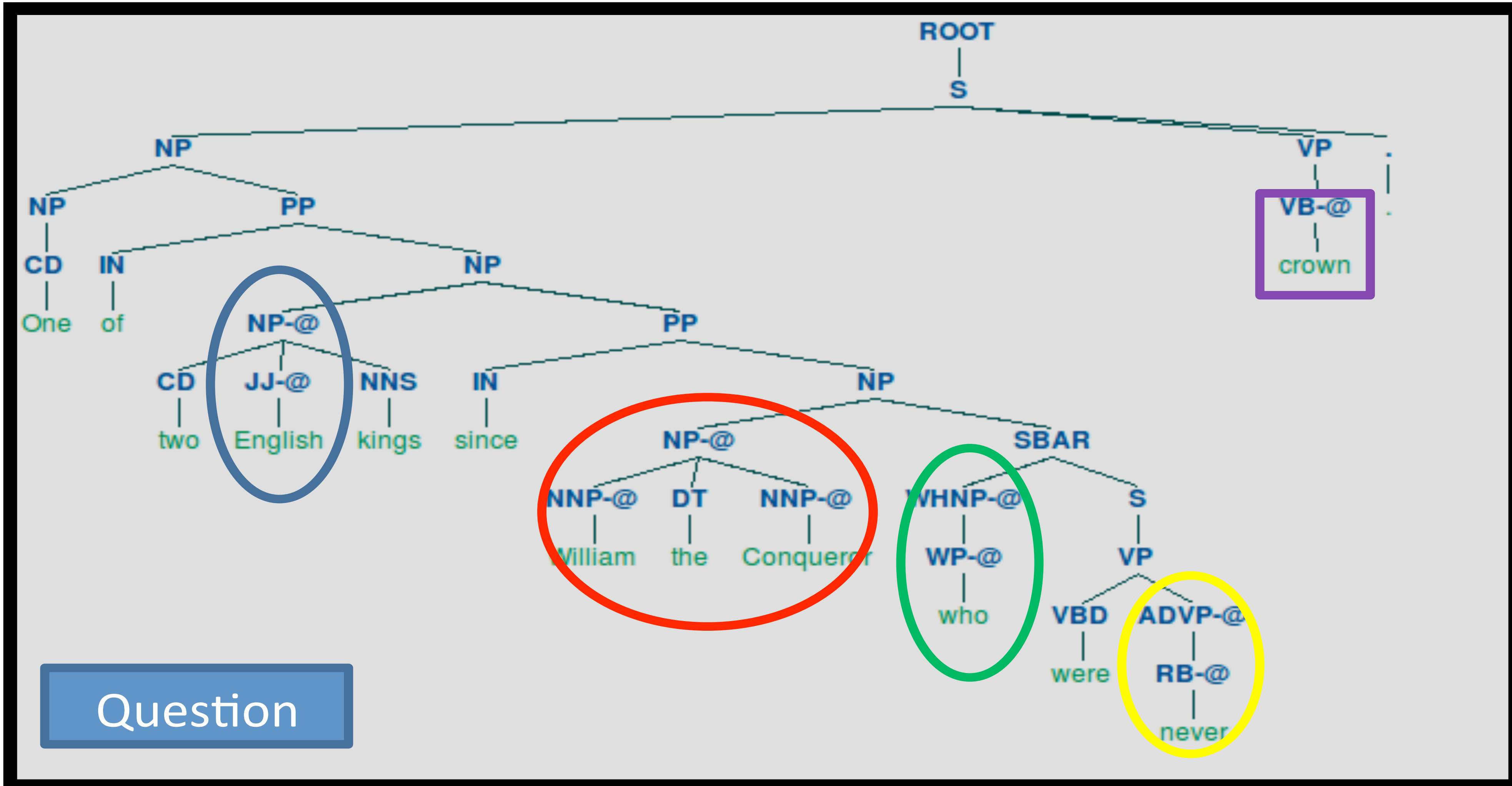

Methodology:

1-Applying lemmatization and stemming in leaves level.

2-Add an anchor to pre-terminal and higher levels if the sub-trees are shared in Q and A.

3-Ignore stop words in matching procedure.




ROOT
S
NP
VP
.
NP
PP
VB-@
CD
IN
NP
crown
One
of
NP-@
PP
.
CD
JJ-@
NNS
IN
NP
two
English
kings
since
NP-@
SBAR
NNP-@
DT
NNP-@
WHNP-@
S
William
the
Conqueror
WP-@
VP
who
VBD
ADVP-@
were
RB-@
never
Question

# Issues

- Very large sentences
- The Jeopardy cues can be constituted by more than one sentence
- The answer is typically composed by several sentences
- Too large structures cause inaccuracies in the similarity and the learning algorithm looses some of its power

# Running example (randomly picked Q/A pair from Answerbag )

**Question**: Is movie theater popcorn vegan?

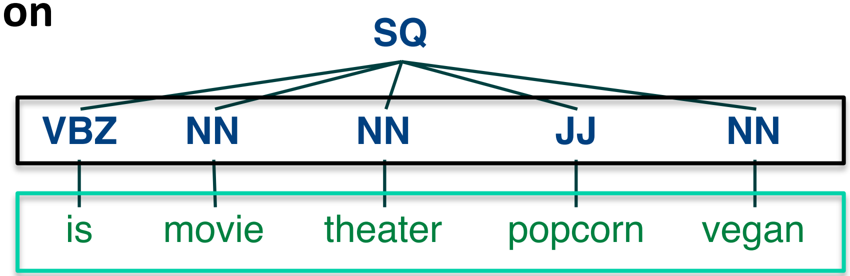
**Answer**:

**(01)** Any movie theater popcorn that includes butter -- and therefore dairy products -- is not vegan.

**(02)** However, the popcorn kernels alone can be considered vegan if popped using canola, coconut or other plant oils which some theaters offer as an alternative to standard popcorn.

# Bag of features: words and part-of-speech tags (use STK on the following strictures)

**Question**




bag of pos tags

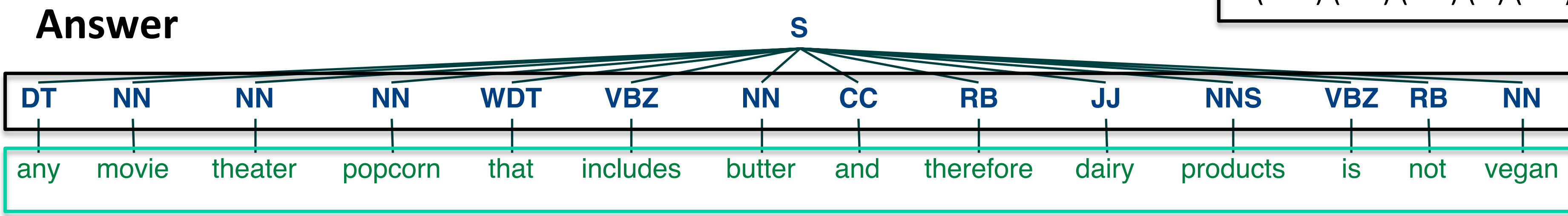
bag of words

and their combination

(is) (movie) (theater) (popcorn) (vegan)

(VBZ) (NN) (NN) (JJ) (NN)

**Answer**

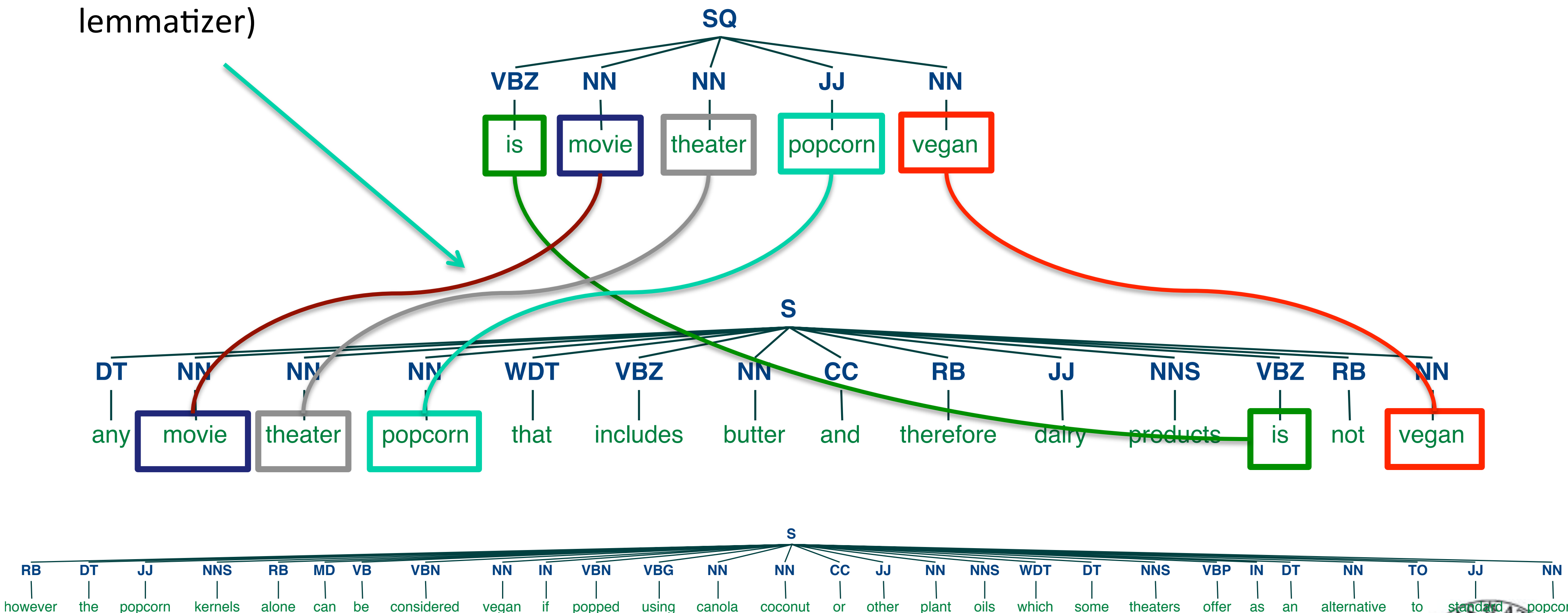



(any) (movie) (theater) (popcorn) (that) (includes) (butter) (and) (therefore) (dairy) (products) (is) (not) (vegan)

(DT) (NN) (NN) (NN) (WDT) (VBZ) (NN) (CC) (RB) (JJ) (NNS) (VBZ) (RB) (NN)

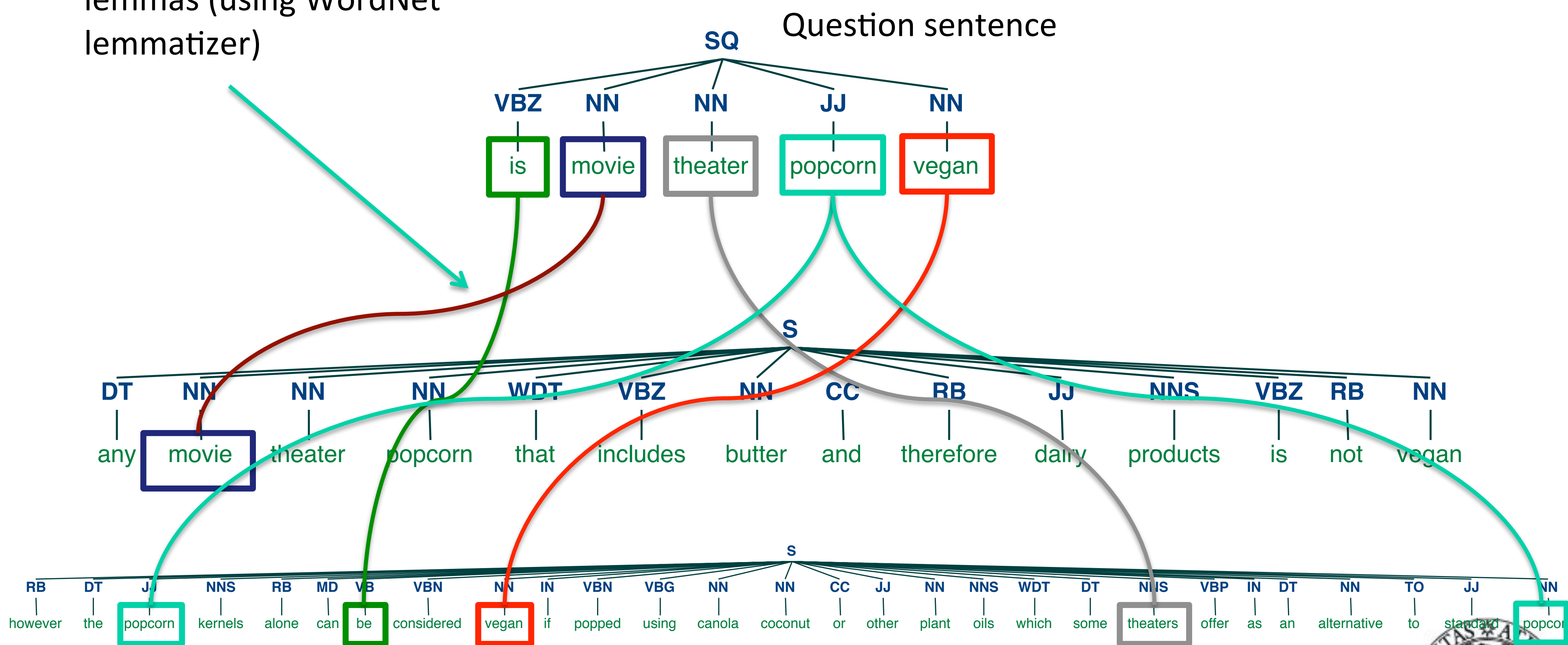
# Linking question with the answer 01

Lexical matching is on word lemmas (using WordNet lemmatizer)



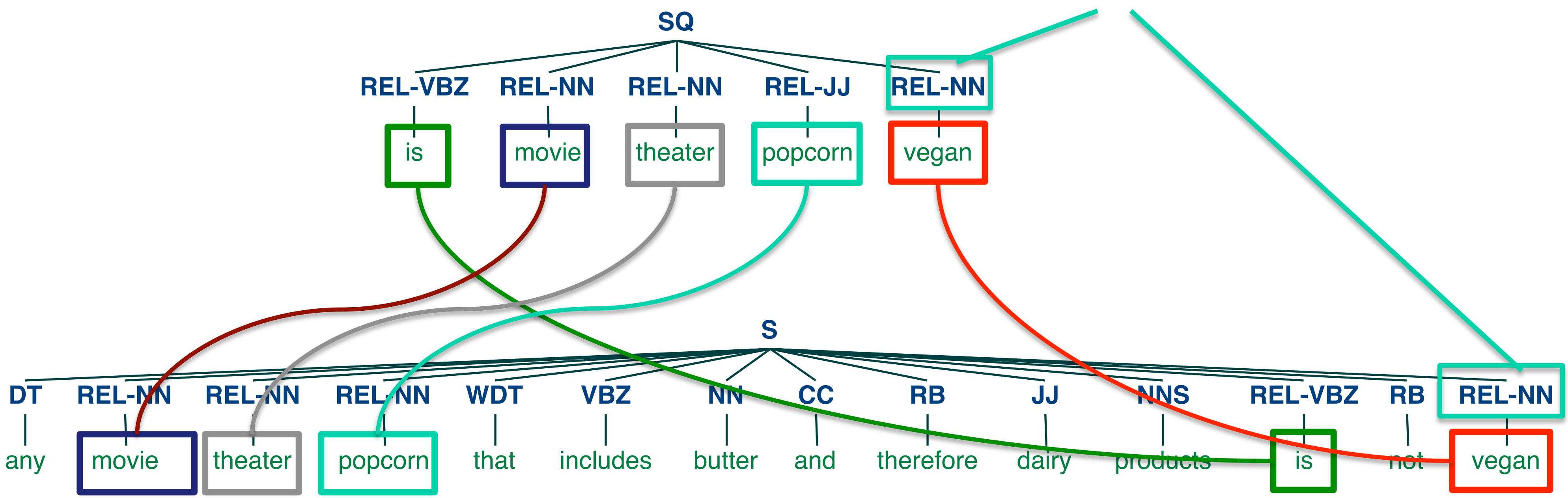
# Linking question with the answer 02

Lexical matching is on word lemmas (using WordNet lemmatizer)



# Linking question with the answer: relational tag

Marking pos tags of the aligned words by a relational tag: "REL"



# Re-ranking Framework

- Start from the most likely set of hypotheses (sometime generated by a basic classifiers)
- These are used to build annotation pairs, $\langle H^i, H^j \rangle$
  - positive instances if $H^i$ is correct and $H^j$ is not correct
- A binary classifier decides if $H^i$ is more probable than $H^j$.
- Each candidate annotation $H^i$ is described by a structural representation
- This way kernels can exploit all dependencies between features and labels

# Kernels for reranking

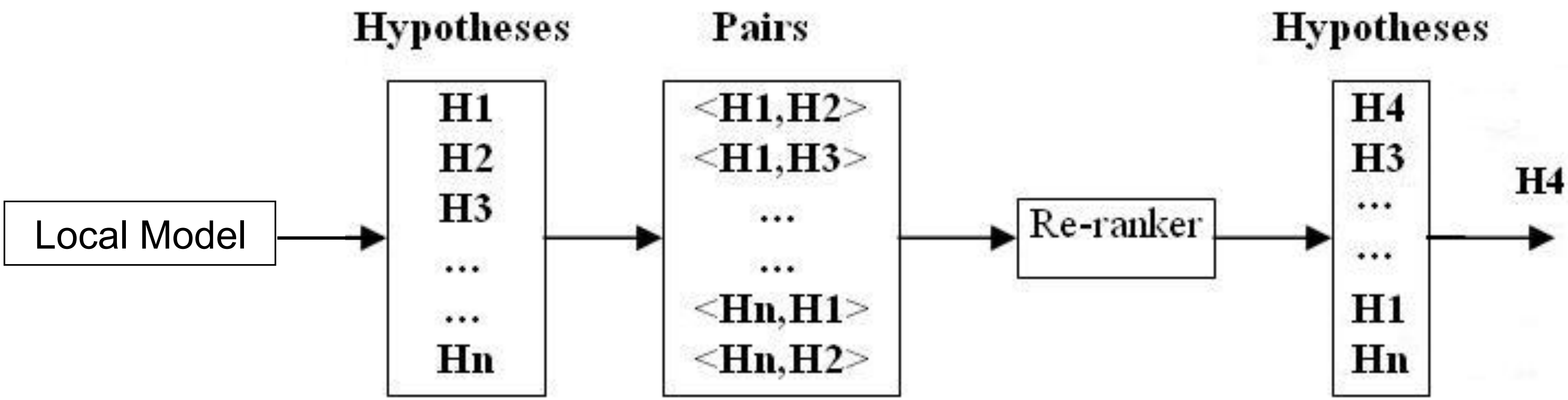
$$P_K(x, y) = \left\langle \phi(x_1) - \phi(x_2), \phi(y_1) - \phi(y_2) \right\rangle =$$

$$P_K(\langle x_1, x_2 \rangle, \langle y_1, y_2 \rangle) = K(x_1, y_1) +$$

$$K(x_2, y_2) - K(x_1, y_2) - K(x_2, y_1),$$

where $K(x_1, y_1) = \mathrm{PTK}\left(q_{x_1}, q_{y_1}\right) + \mathrm{PTK}\left(a_{x_1}, a_{y_1}\right)$

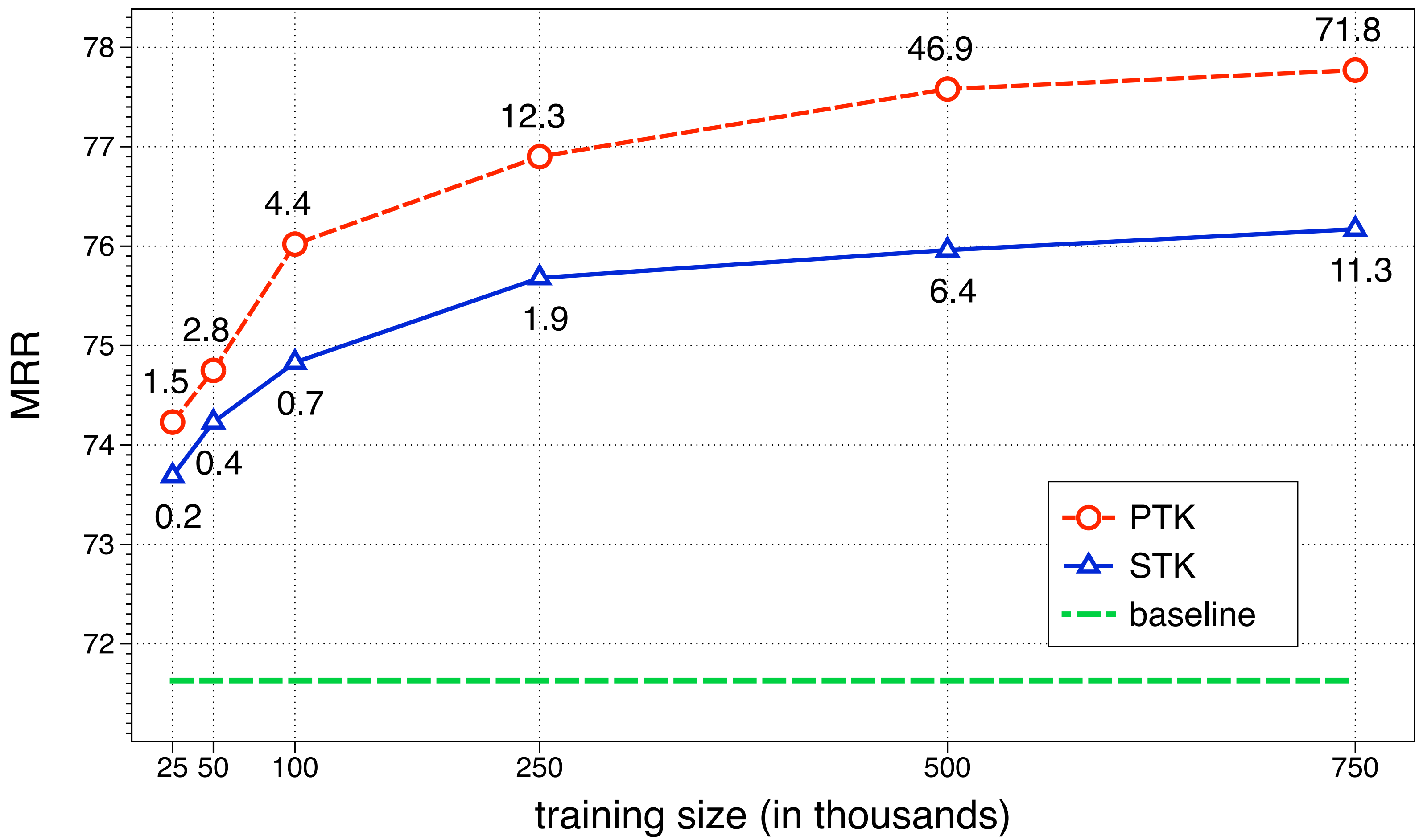
# Re-ranking framework



# Answerbag data

- www.answerbag.com: professional question answer interactions
- Divided in 30 categories, Art, education, culture, …
- 180,000 question-answer pairs

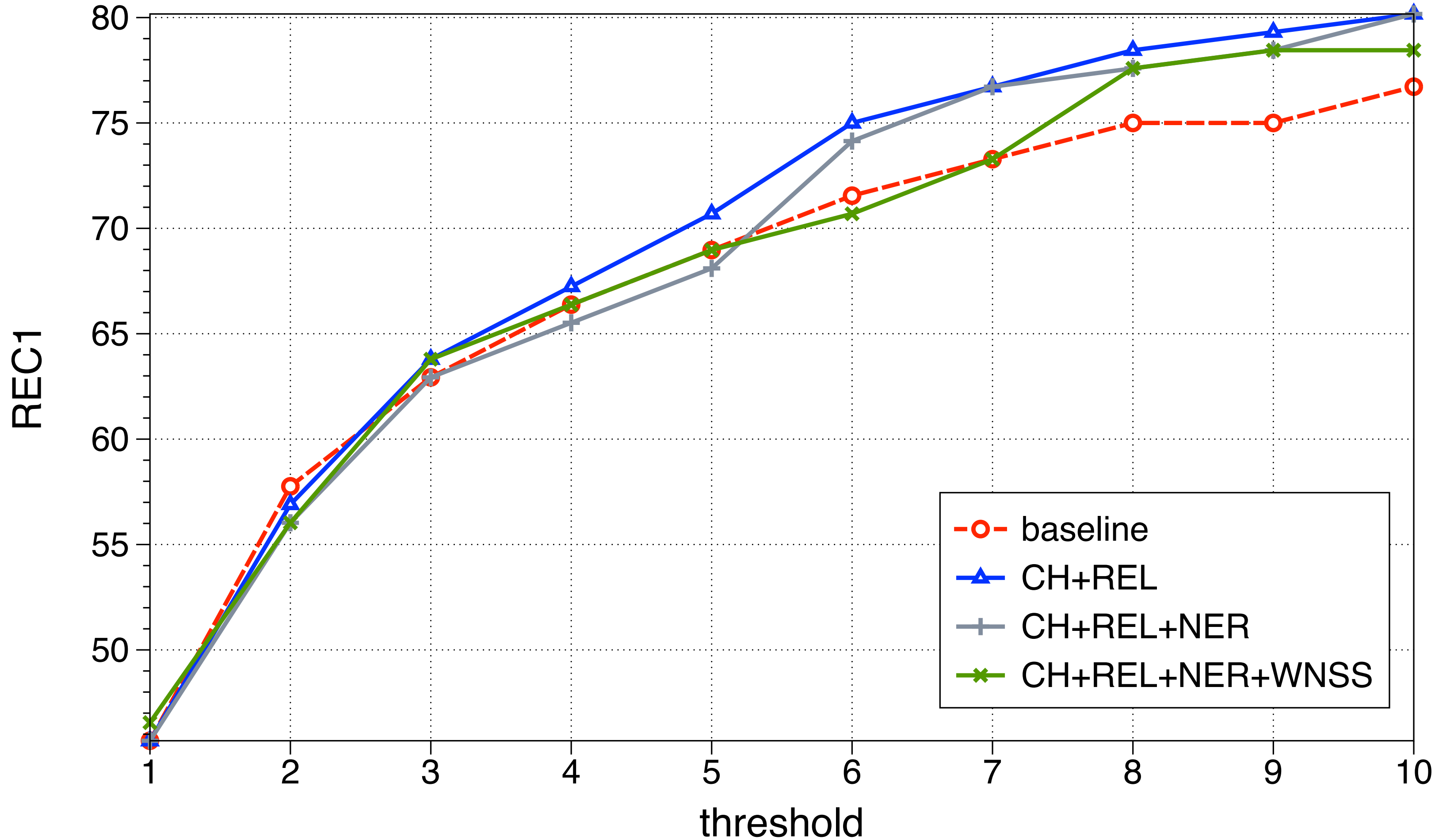
# Learning Curve-Answerbag



# Jeopardy data (T9)

- Total number of questions: 517
- 50+ candidate answer passages per question
- Questions with at least one correct answer: 375
- Use only questions with at least one correct answer
- Each relevant passage is paired with each irrelevant
- Split the data:
  - train 70% (259 questions) -> 63361 examples for re-ranker
  - test 30% (116 question) -> 5706 examples for re-ranker

# Jeopardy! data



# SEMANTIC ROLE LABELING

# Example on Predicate Argument Classification

- In an event:
  - target words describe relation among different entities
  - the participants are often seen as predicate's arguments.
- Example:

  Paul gives a talk in Rome

# Example on Predicate Argument Classification

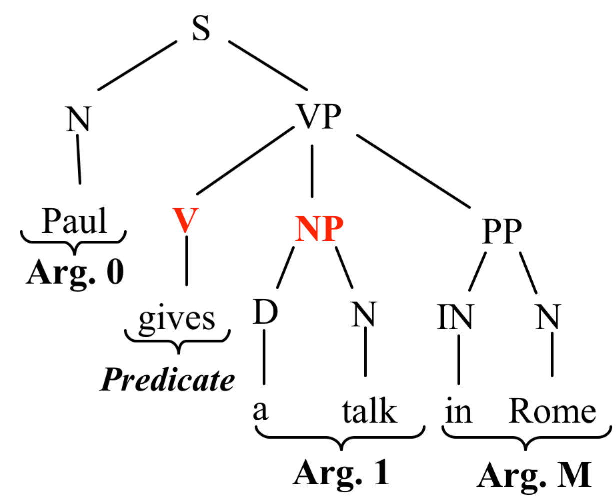
- In an event:
  - target words describe relation among different entities
  - the participants are often seen as predicate's arguments.
- Example:

  [ $_{Arg0}$ Paul] [ $_{predicate}$ gives ] [ $_{Arg1}$ a talk] [ $_{ArgM}$ in Rome]

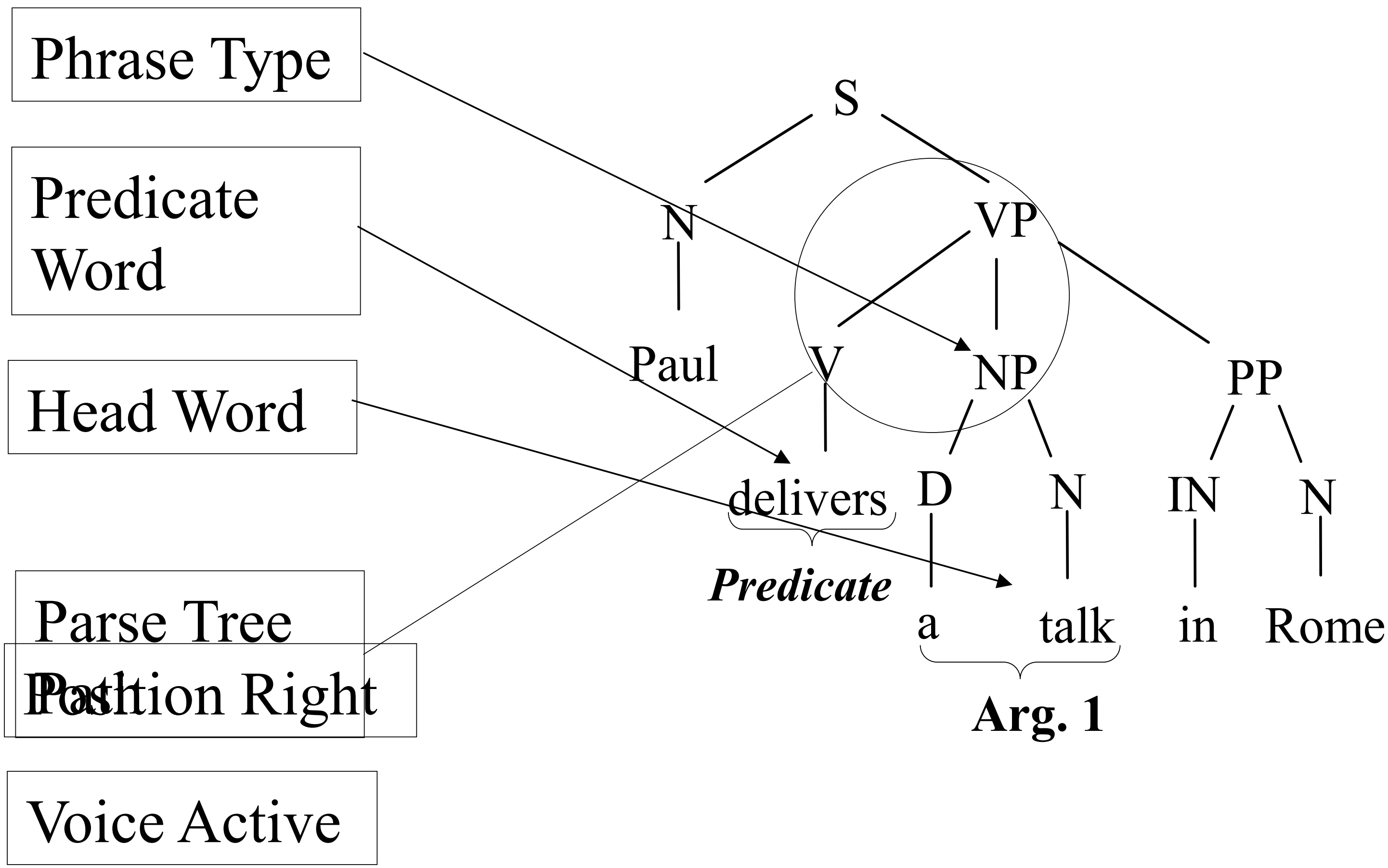
# Predicate-Argument Feature Representation

Given a sentence, a predicate *p*:

1. Derive the sentence parse tree
2. For each node pair <$N_p$,$N_x$>
   a. Extract a feature representation set *F*
   b. If $N_x$ exactly covers the Arg-$i$, *F* is one of its positive examples
   c. *F* is a negative example otherwise



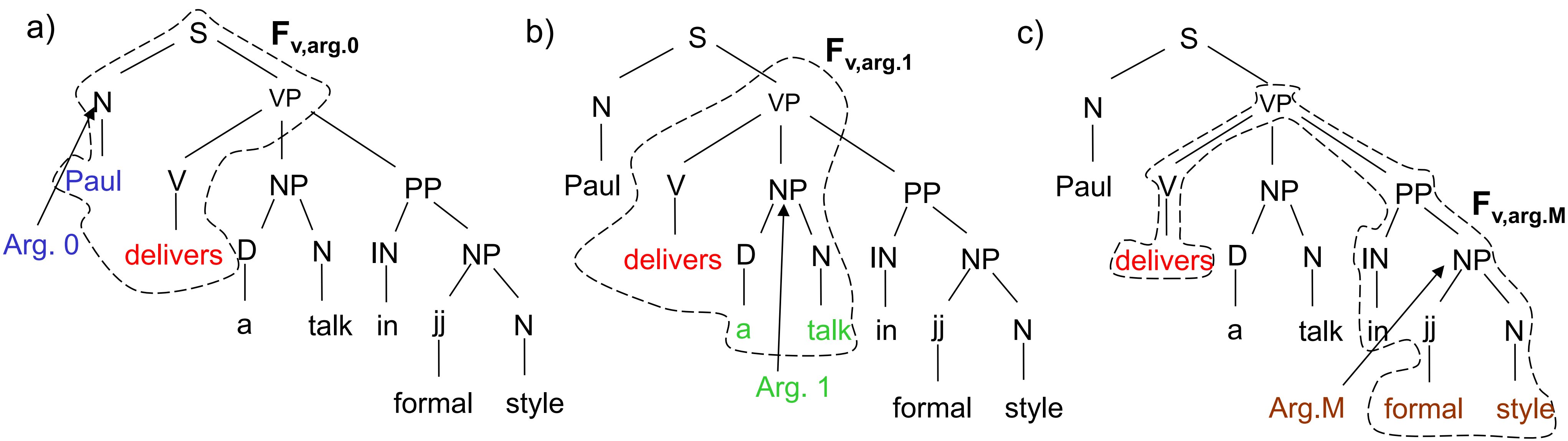
# Vector Representation for the linear kernel



# PAT Kernel [Moschitti, ACL 2004]

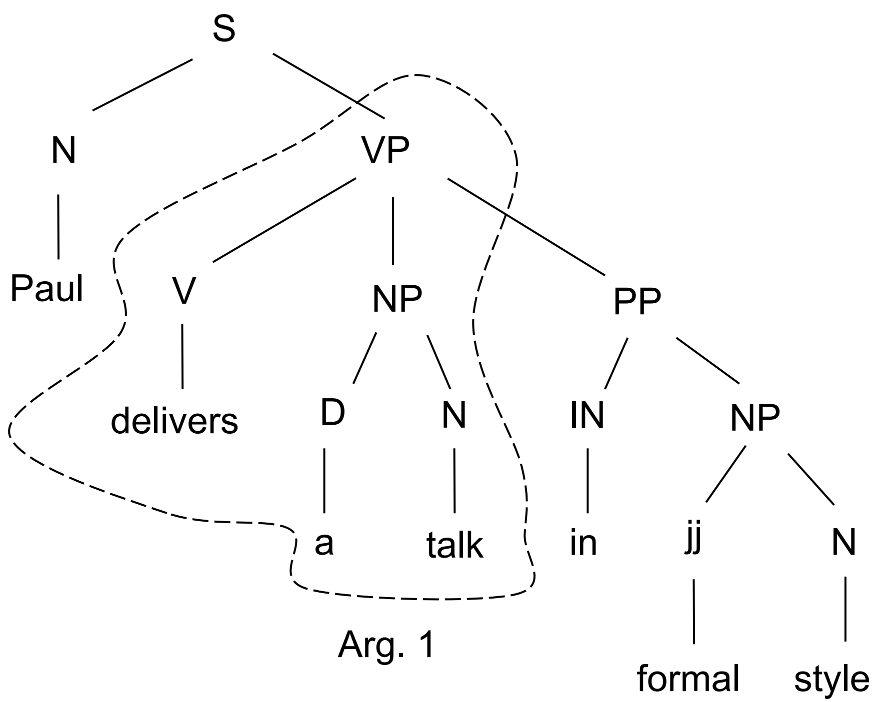
- Given the sentence:

[ $_{Arg0}$ Paul] [ $_{predicate}$ delivers] [ $_{Arg1}$ a talk] [ $_{ArgM}$ in formal Style]

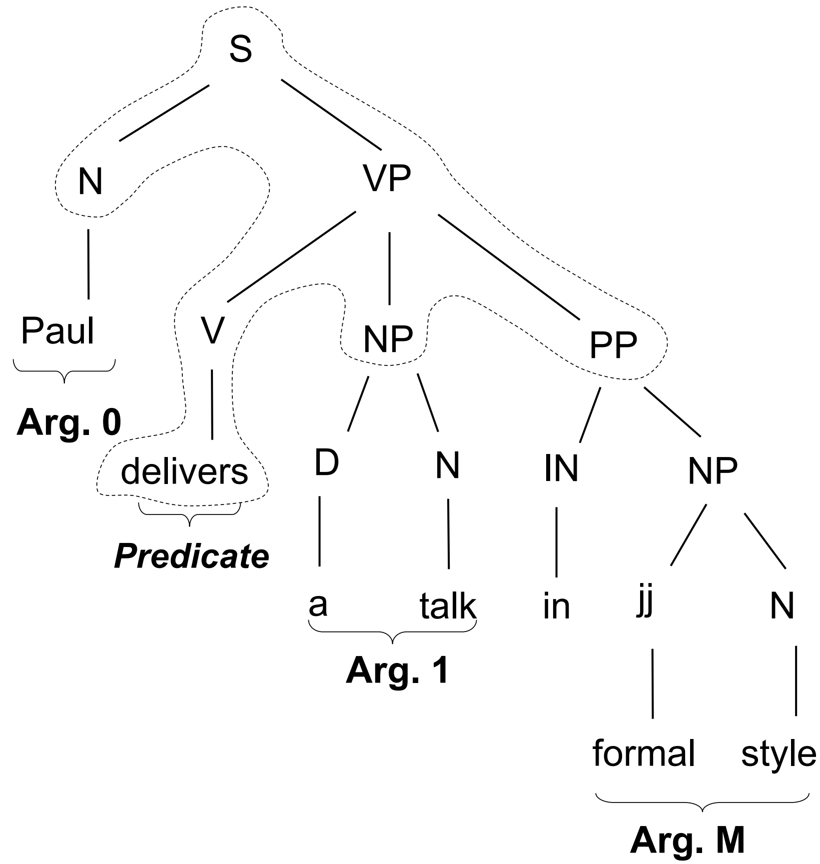



- These are Semantic Structures

# In other words we consider…



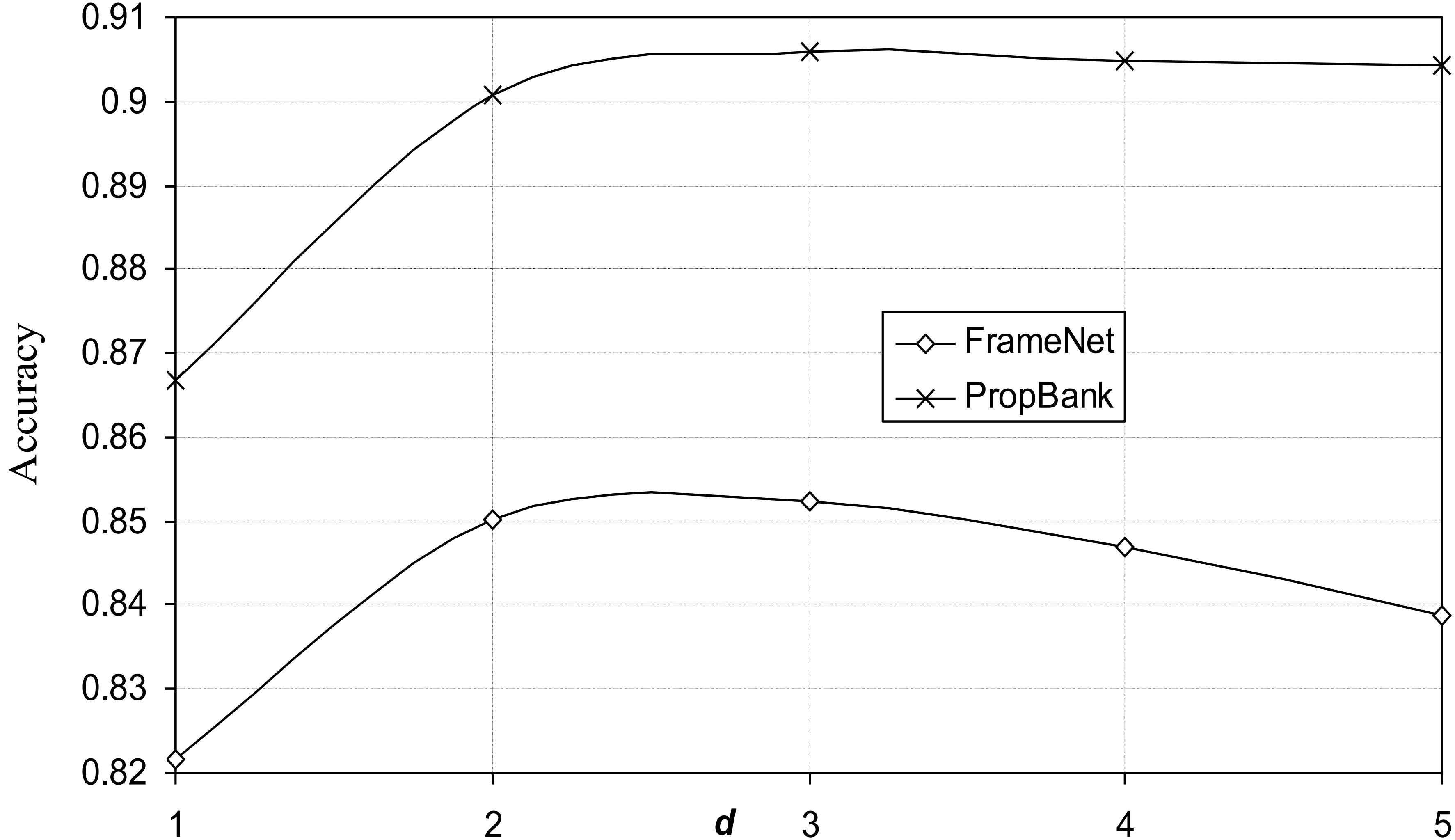
# Sub-Categorization Kernel (SCF) [Moschitti, ACL 2004]



# Experiments on Gold Standard Trees

- PropBank and PennTree bank
  - about 53,700 sentences
  - Sections from 2 to 21 train., 23 test., 1 and 22 dev.
  - Arguments from Arg0 to Arg5, ArgA and ArgM for a total of 122,774 and 7,359
- FrameNet and Collins' automatic trees
  - 24,558 sentences from the 40 frames of Senseval 3
  - 18 roles (same names are mapped together)
  - Only verbs
  - 70% for training and 30% for testing

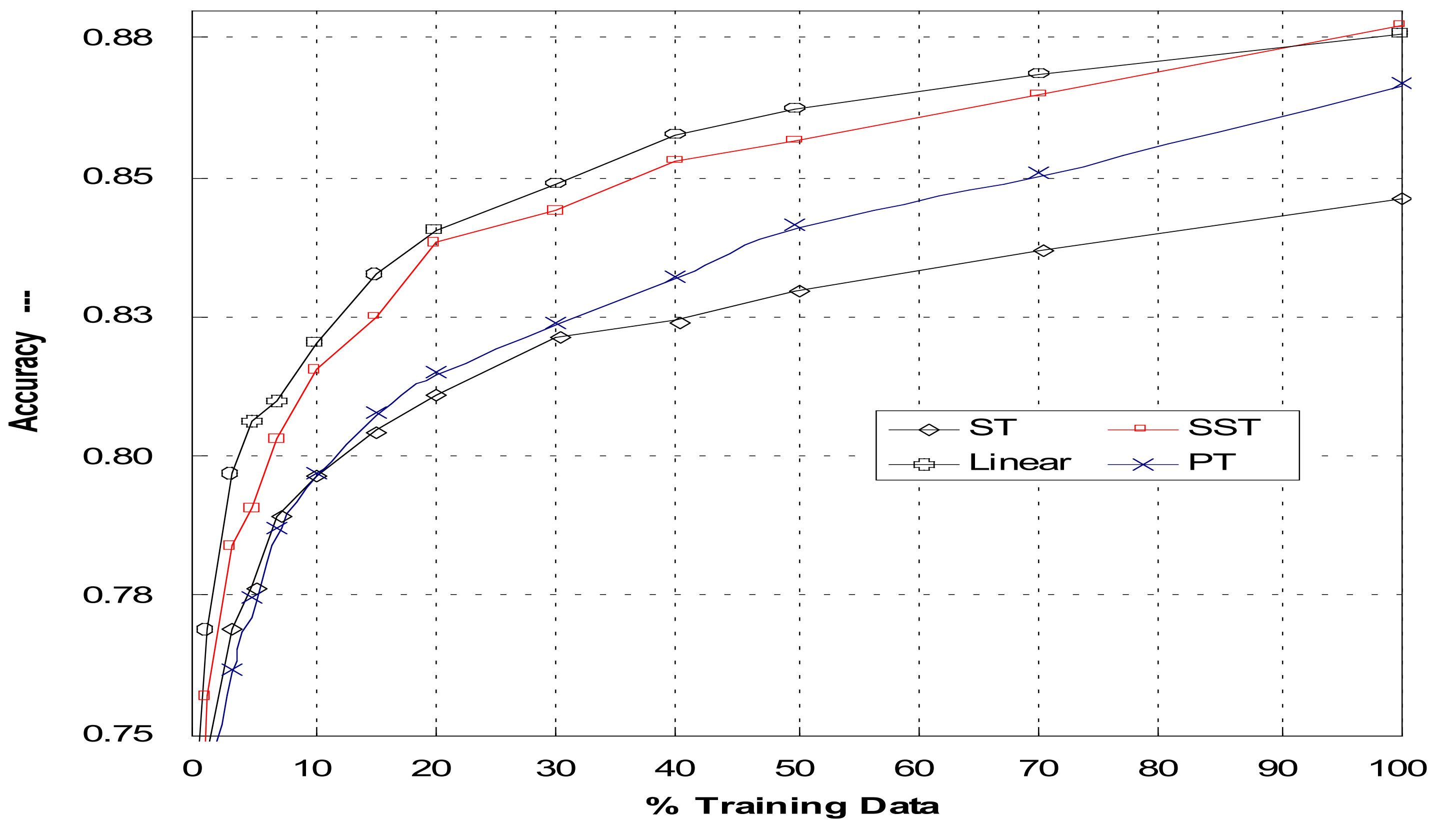
# Argument Classification with Poly Kernel



# PropBank Results

| Args | P3 | PAT | PAT+P | PAT×P | SCF+P | SCF×P |
|---|---|---|---|---|---|---|
| Arg0 | 90.8 | 88.3 | 92.6 | 90.5 | 94.6 | 94.7 |
| Arg1 | 91.1 | 87.4 | 91.9 | 91.2 | 92.9 | 94.1 |
| Arg2 | 80.0 | 68.5 | 77.5 | 74.7 | 77.4 | 82.0 |
| Arg3 | 57.9 | 56.5 | 55.6 | 49.7 | 56.2 | 56.4 |
| Arg4 | 70.5 | 68.7 | 71.2 | 62.7 | 69.6 | 71.1 |
| ArgM | 95.4 | 94.1 | 96.2 | 96.2 | 96.1 | 96.3 |
| **Global Accuracy** | **90.5** | **88.7** | **91.3** | **90.4** | **92.4** | **93.2** |

# Argument Classification on PAT using different Tree Fragment Extractor

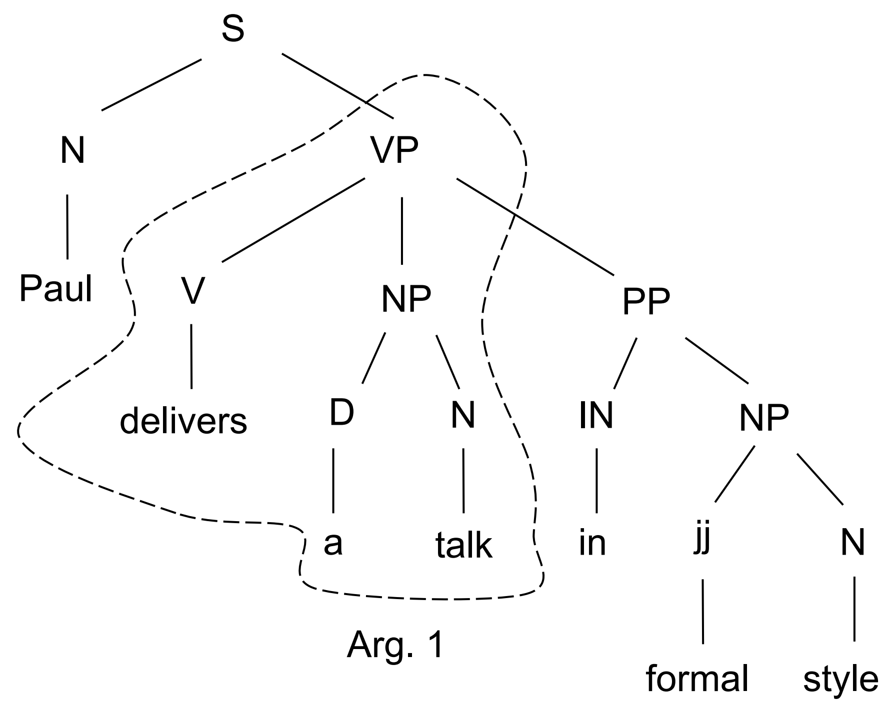


# FrameNet Results

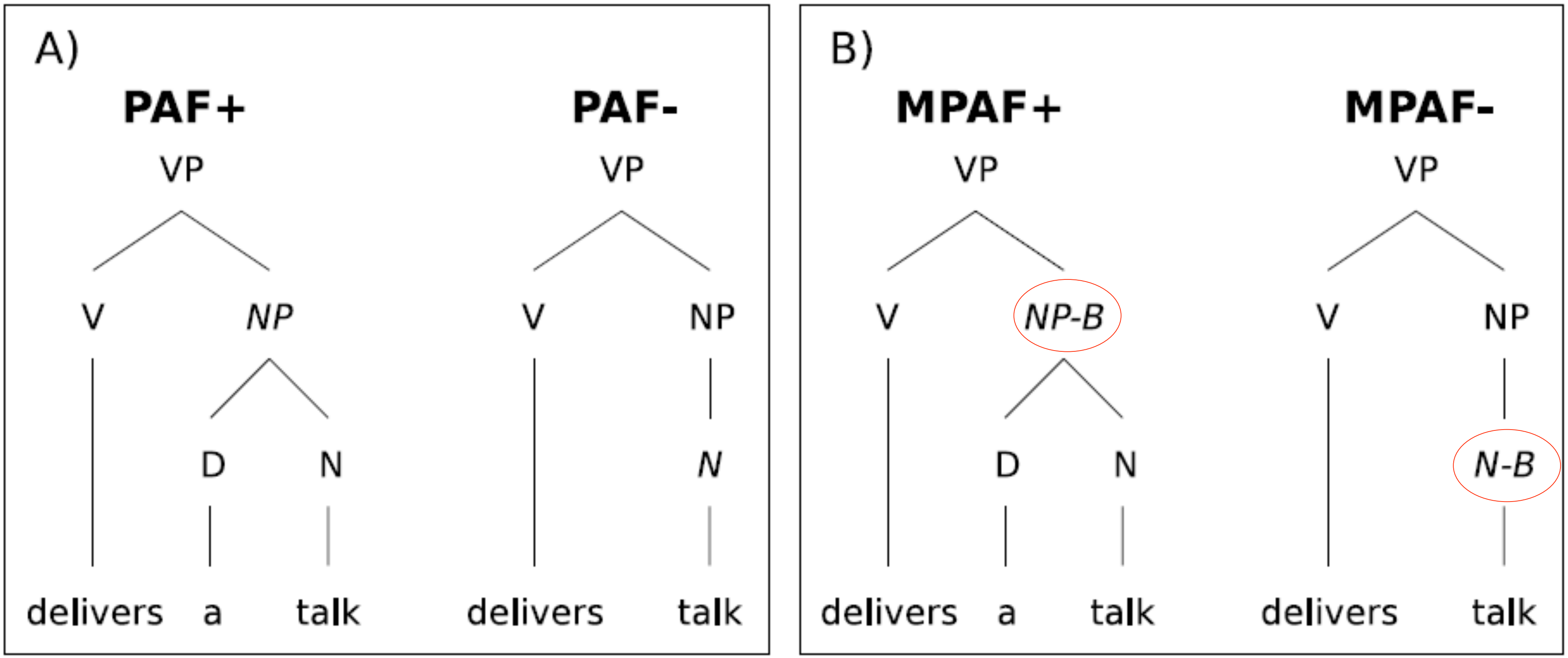
| Roles | P3 | PAF | PAF+P | PAF×P | SCF+P | SCF×P |
|---|---|---|---|---|---|---|
| agent | 92.0 | 88.5 | 91.7 | 91.3 | 93.1 | 93.9 |
| cause | 59.7 | 16.1 | 41.6 | 27.7 | 42.6 | 57.3 |
| degree | 74.9 | 68.6 | 71.4 | 57.8 | 68.5 | 60.9 |
| depictive | 52.6 | 29.7 | 51.0 | 28.6 | 46.8 | 37.6 |
| duration | 45.8 | 52.1 | 40.9 | 29.0 | 31.8 | 41.8 |
| goal | 85.9 | 78.6 | 85.3 | 82.8 | 84.0 | 85.3 |
| instrument | 67.9 | 46.8 | 62.8 | 55.8 | 59.6 | 64.1 |
| manner | 81.0 | 81.9 | 81.2 | 78.6 | 77.8 | 77.8 |
| Global Acc. (18 roles) | 85.2 | 79.5 | 84.6 | 81.6 | 83.8 | 84.2 |

- ProbBank arguments vs. Semantic Roles

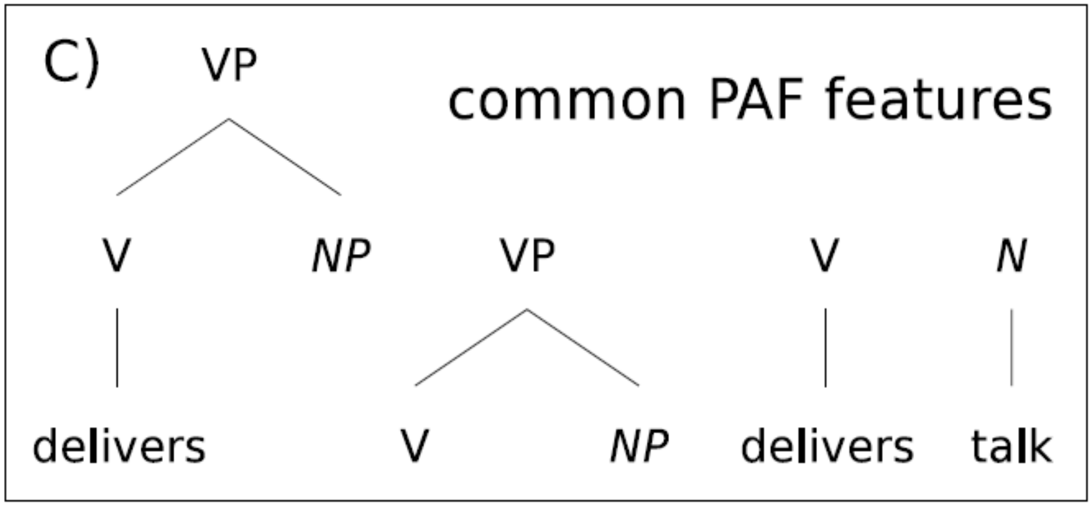
# Boundary Detection



# Improvement by Marking Boundary nodes



# Node Marking Effect




# Experiments

- PropBank and PennTree bank
  - about 53,700 sentences
  - Charniak trees from CoNLL 2005
- Boundary detection:
  - Section 2 training
  - Section 24 testing
  - PAF and MPAF

# Number of examples/nodes of Section 2

| | Section 2 | | | Section 24 | | |
|---|---|---|---|---|---|---|
| Nodes | pos | neg | tot | pos | neg | tot |
| Internal | 11,847 | 71,126 | 82,973 | 7,525 | 50,123 | 57,648 |
| Pre-terminal | 894 | 114,052 | 114,946 | 709 | 80,366 | 81,075 |
| Both | 12,741 | 185,178 | 197,919 | 8,234 | 130,489 | 138,723 |

# Predicate Argument Feature (PAF) vs. Marked PAF (MPAF) [Moschitti et al, CLJ 2008]

State-of-the-art:
- Boundary detection PropBank
- Arabic SRL (Diab et al, 2008)

| Tagging strategy | $CPU_{time}$ | F1 |
|---|---|---|
| PAF | 5,179.18 | 75.24 |
| MPAF | 3,131.56 | 82.07 |

# Results on FrameNet SRL
## [Coppola and Moschitti, LREC 2010]

- 135,293 annotated and parsed sentences.
- 782 different frames (including split per pos-tag)
- 90% of training data for BD and BC 121,798 sentences
- 10% of testing data (1,345 sentences)

| Enhanced PK+TK | | | |
|---|---|---|---|
| Eval Setting | $P$ | $R$ | $F_1$ |
| BD (nodes) | 1.0 | .732 | .847 |
| BD (words) | .963 | .702 | .813 |
| BD+RC (nodes) | .784 | .571 | .661 |
| BD+RC (words) | .747 | .545 | .630 |

# Experiments on Luna Corpus
## [Coppola at al, SLT 2008]

- BD and RC over 50 Human-Human dialogs
  - 1677 target words spanning 162 different frames
  - manually-corrected syntactic trees
  - Training 90% data and testing on remaining 10%

State-of-the-art:
- FrameNet (difficult comparison)
- First system on SLU

| Evaluation Stage | Precision | Recall | F1 |
|---|---|---|---|
| Boundary Detection | 0.905 | 0.873 | 0.889 |
| Boundary Detection + Role Classification | 0.774 | 0.747 | 0.760 |

- Automatic SRL viable for Spoken Dialog Data.

# RELATION EXTRACTION

# The Extraction Problem

Last Wednesday, Eric Schmidt, the CEO of Google, defended the search engine's cooperation with Chinese censorship as he announced the creation of a research center in Beijing.

EMPLOYMENT

CEO ↔ Google

LOCATED

research center ↔ Beijing

Given a text with some available entities, how to recognize relations ?

# Relation Extraction: The task

- Task definition: to label the semantic relation between pairs of entities in a sentence
  - The **governor** from **Connecticut**

M1
type: PER

M2
type: LOC

M := Entity Mention

  - Is there a relation between M1 and M2?
    If, so what kind of relation?

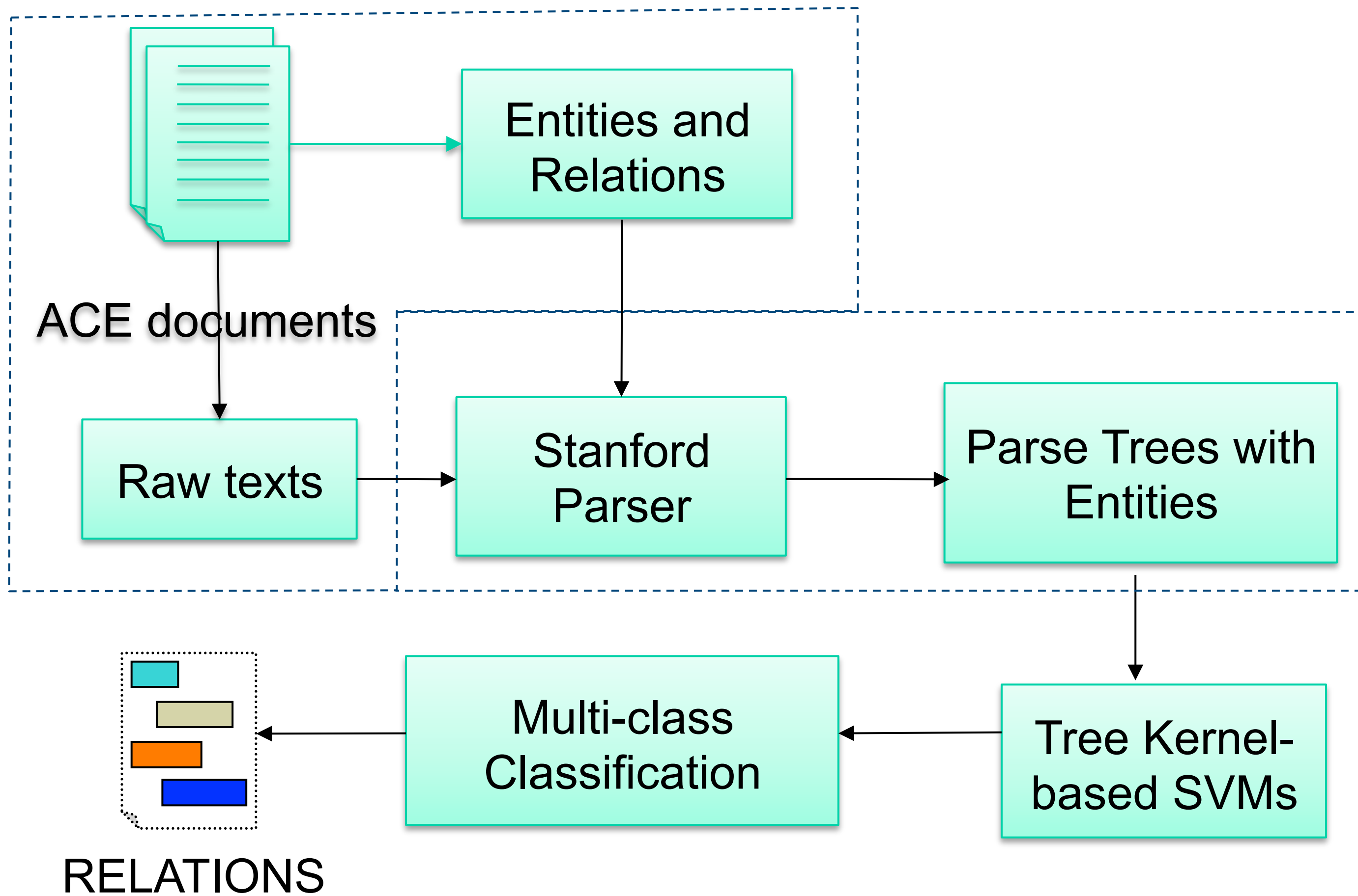
# Relation Extraction defined in ACE

- Major relation types (from ACE 2004)

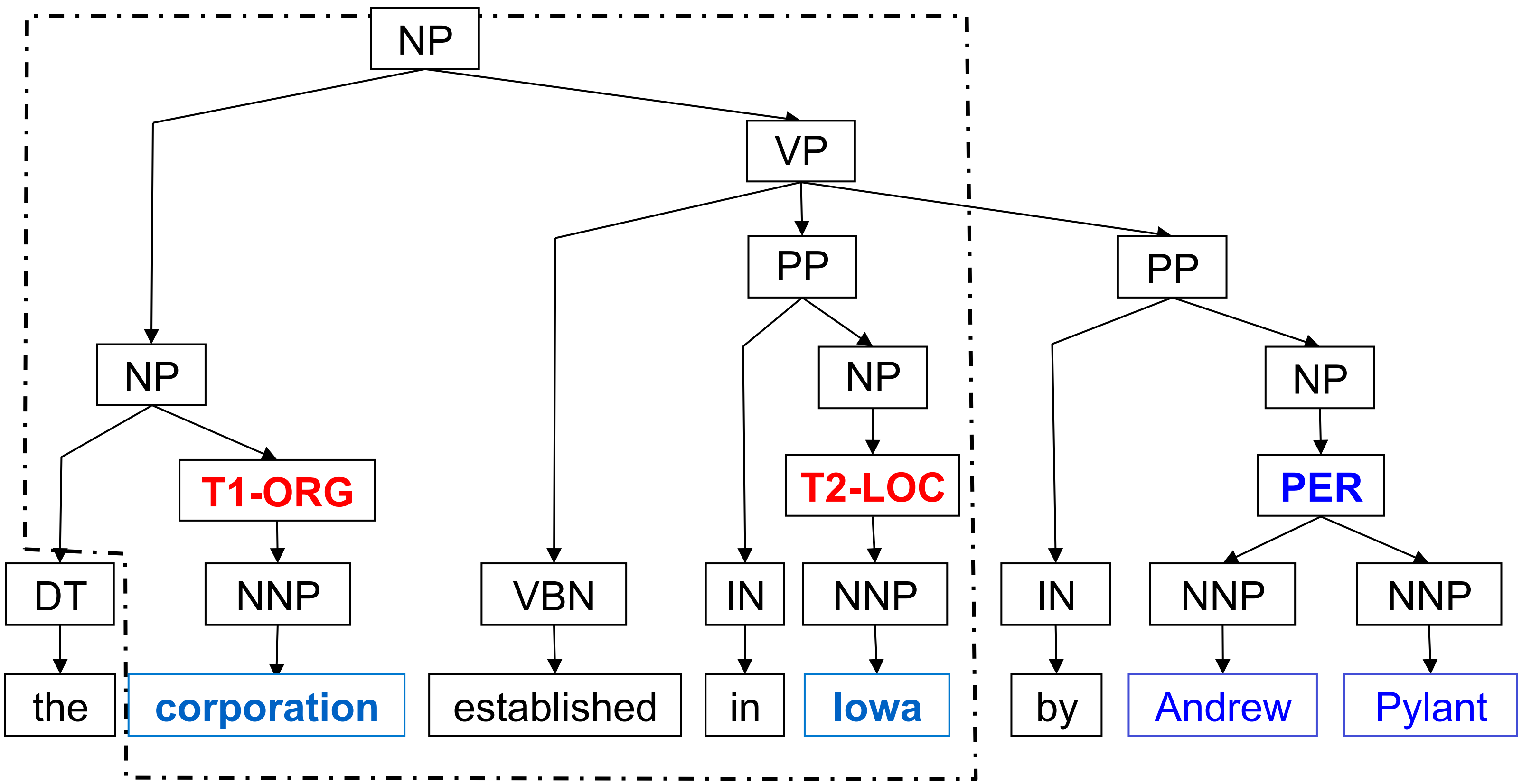
| Type | Definition | Example |
|---|---|---|
| EMP-ORG | **Employment** | *US president* |
| PHYS | **Located, near, part-whole** | *a military base in Germany* |
| GPE-AFF | **Affiliation** | *U.S. businessman* |
| PER-SOC | **Social** | *a spokesman for the senator* |
| DISC | **Discourse** | *each of whom* |
| ART | **User, owner, inventor …** | *US helicopters* |
| OTHER-AFF | **Ethnic, ideology …** | *Cuban-American people* |

- Entity types: PER, ORG, LOC, GPE, FAC, VEH, WEA

# System Description (Nguyen et al, 2009)



# Relation Representation
## (Moschitti 2004;Zhang et al. 2006)




- The Path-enclosed tree captures the “PHYSICAL.LOCATED” relation between “**corporation**” and “**Iowa**”

# Comparison

| | Method | Data | P (%) | R (%) | F1 (%) |
|---|---|---|---|---|---|
| Zhang et al. (2006) | Composite Kernel (linear) with Context-Free Parse Tree | ACE 2004 | 73.5 | 67.0 | 70.1 |
| Ours | Composite Kernel (linear) with Context-Free Parse Tree | ACE 2004 | 69.6 | 68.2 | 69.2 |

Both use the Path-Enclosed Tree for Relation Representation

# Several Combination Kernels
## [Vien et al, EMNLP 2009]

$$CK_1 = \alpha \cdot K_P + (1-\alpha) \cdot K_x$$

$$CK_2 = \alpha \cdot K_P + (1-\alpha) \cdot (K_{SST} + K_{PTK})$$

$$CK_3 = \alpha \cdot K_{SST} + (1-\alpha) \cdot (K_P + K_{PTK})$$

$$CK_4 = K_{PTK-DW} + K_{PTK-GR}$$

$$CK_5 = \alpha \cdot K_P + (1-\alpha) \cdot (K_{PTK-DW} + K_{PTK-GR})$$

$$SSK = \sum_{i=1,..,6} SK_i$$

$$CSK = \alpha \cdot K_P + (1-\alpha) \cdot (K_{SST} + SSK)$$

| Kernel | P | R | F |
|---|---|---|---|
| $\mathbf{CK_1}$ | **69.5** | **68.3** | **68.9** |
| $SK_1$ | 72.0 | 52.8 | 61.0 |
| $SK_2$ | 61.7 | 60.0 | 60.8 |
| $SK_3$ | 62.6 | 60.7 | 61.6 |
| $SK_4$ | 73.1 | 50.3 | 59.7 |
| $SK_5$ | 59.0 | 60.7 | 59.8 |
| $SK_6$ | 57.7 | 61.8 | 59.7 |
| $\mathbf{SK_3 + SK_4}$ | [illegible] | **63.4** | **68.8** |
| $SK_3 + SK_6$ | 66.8 | 65.1 | 65.9 |
| $\mathbf{SSK = \sum_i SK_i}$ | **73.8** | **66.2** | **69.8** |
| **SST Kernel + SSK** | **75.6** | **66.6** | **70.8** |
| $\mathbf{CK_1 + SSK}$ | **76.6** | **67.0** | **71.5** |
| *(Zhou et al., 2007)* *$CK_1$ with Heuristics* | *82.2* | *70.2* | *75.8* |

State-of-the-art

# COREFERENCE RESOLUTION

# Syntactic Tree feature

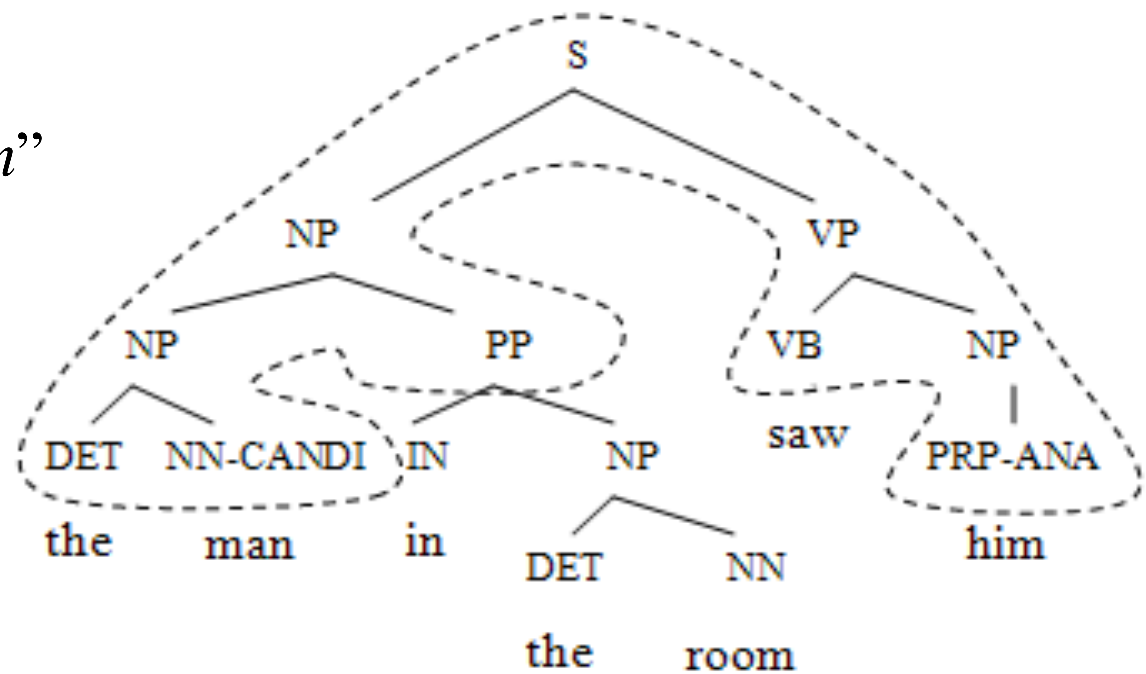
- Subtree that covers both anaphor and antecedent candidate

⇒ syntactic relations between anaphor & candidate (subject, object, c-commanding, predicate structure)

- Include the nodes in path between anaphor and candidate, as well as their first_level children

–"*the man* in the room saw *him*"

– inst("the man", "him")



# Context Sequence Feature

- A word sequence representing the mention expression and its context
  - Create a sequence for a mention

– "Even so, **Bill Gates** says that he just doesn't understand our infatuation with thin client versions of Word "

– (so)(,) (**Bill**)(**Gates**)(says)(that)

# Composite Kernel

- different kernels for different features
  - Poly Kernel for baseline flat features
  - Tree Kernel for syntax trees
  - Sequence Kernel for word sequences
- A composite kernel for all kinds of features
- Composite Kernel = TK*PolyK+PolyK+SK

# Results for pronoun resolution [Vesley et al, Coling 2008]

| | MUC-6 | | | ACE-02-BNews | | |
|---|---|---|---|---|---|---|
| | R | P | F | R | P | F |
| All attribute value features | 64.3 | 63.1 | 63.7 | 58.9 | 68.1 | 63.1 |
| + Syntactic Tree + Word Sequence | 65.2 | 80.1 | **71.9** | 65.6 | 69.7 | **67.6** |

State-of-the-art

# Results for over-all coreference Resolution using SVMs

| | MUC-6 | | | ACE02-BNews | | |
|---|---|---|---|---|---|---|
| | R | P | F | R | P | F |
| BaseFeature SVMs | 61.5 | 67.2 | 64.2 | 54.8 | 66.1 | 59.9 |
| BaseFeature + Syntax Tree | 63.4 | 67.5 | 65.4 | 56.6 | 66.0 | 60.9 |
| BaseFeature +SyntaxTree + Word Sequences | 64.4 | 67.8 | 66.0 | 57.1 | 65.4 | 61.0 |
| All Sources of Knowledge | 60.1 | 76.2 | 67.2 | 60.0 | 65.4 | 63.0 |

# RECOGNIZING TEXTUAL ENTAILMENT

# Target Problem

learning **textual entailment recognition rules**
from annotated examples

… the textual entailment recognition task:

determine whether or not a text T implies a hypothesis H

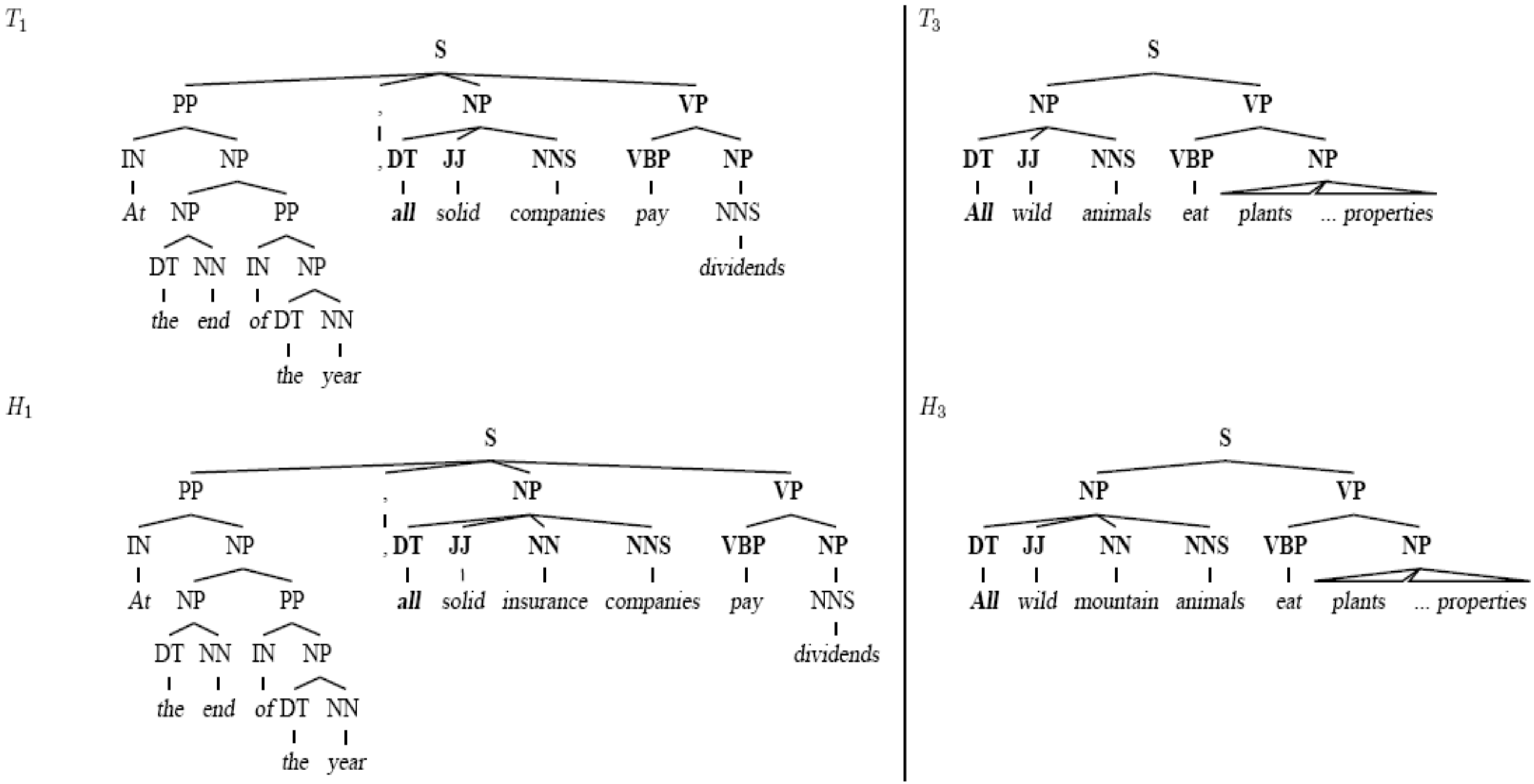
$T_1 \Rightarrow H_1$

| | |
|---|---|
| $T_1$ | *"At the end of the year, all solid companies pay dividends."* |
| $H_1$ | *"At the end of the year, all solid insurance companies pay dividends."* |

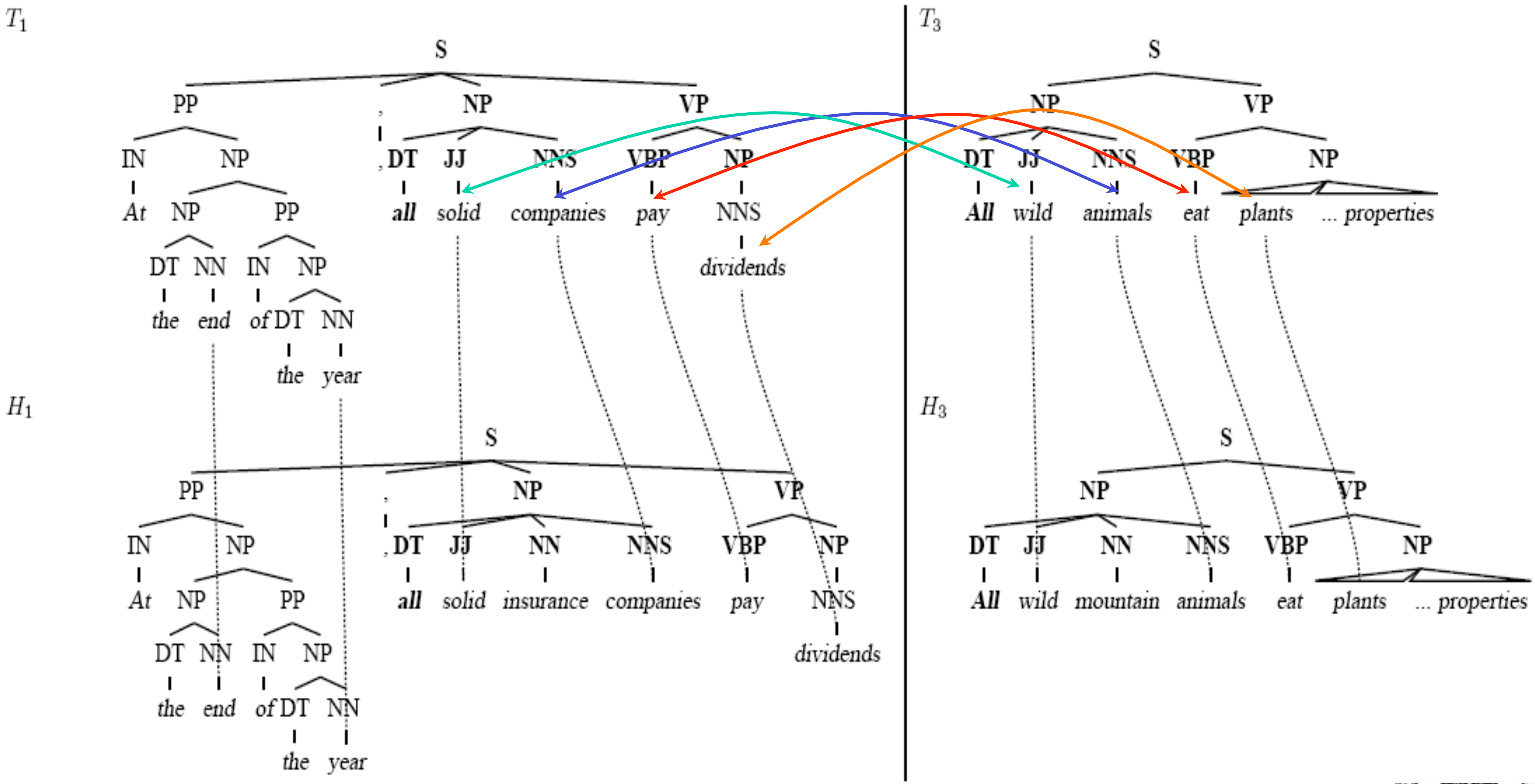
***"Traditional" machine learning approaches:***

similarity-based methods → distance in feature spaces

# Determine Intra-pair links



# Determine cross pair links



# Our Model (Zanzotto and Moschitti, ACL2006)

Defining a similarity between pairs based on:

$$K_{ent}((T',H'),(T'',H''))=K_I((T',H'),(T'',H''))+K_S((T',H'),(T'',H''))$$

- Intra-pair similarity

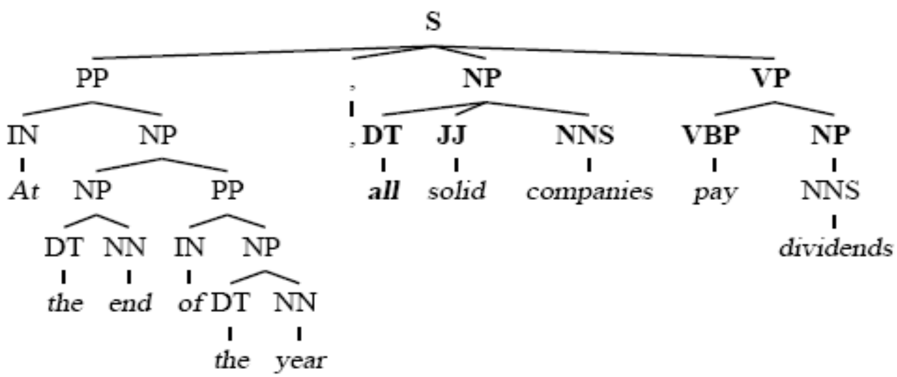
$$K_I((T',H'),(T'',H''))=s(T',H')\times s(T'',H'')$$

- Cross-pair similarity

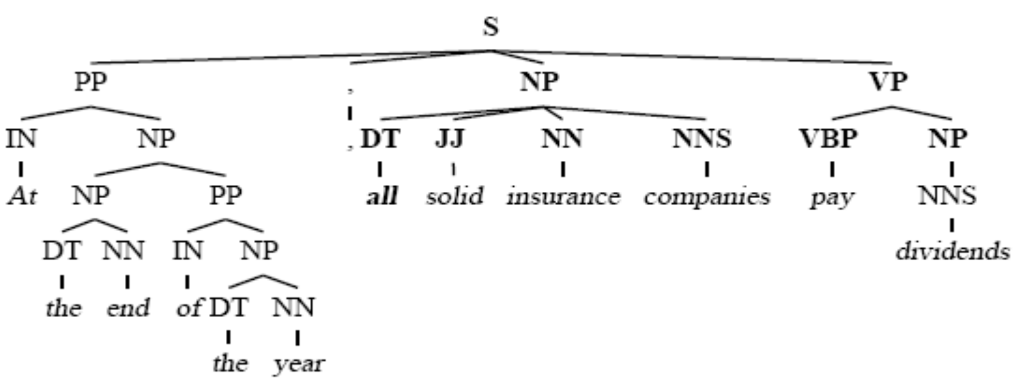
$$K_S((T',H'),(T'',H''))\approx K_T(T',T'')+K_T(H',H'')$$

# Our Model: an example

$T_1$



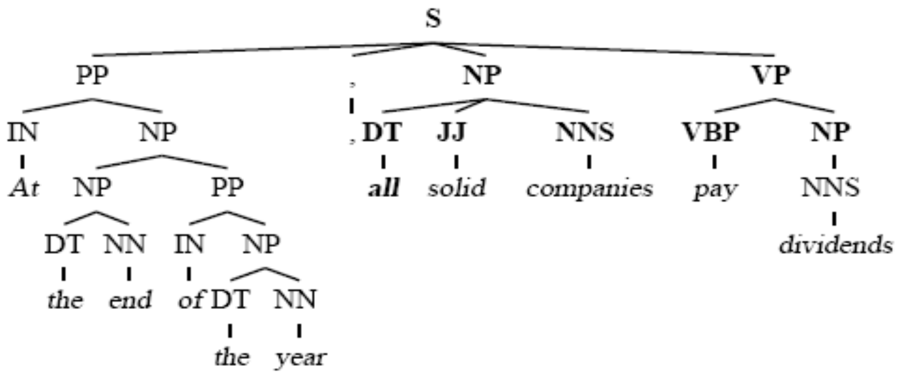

$H_1$



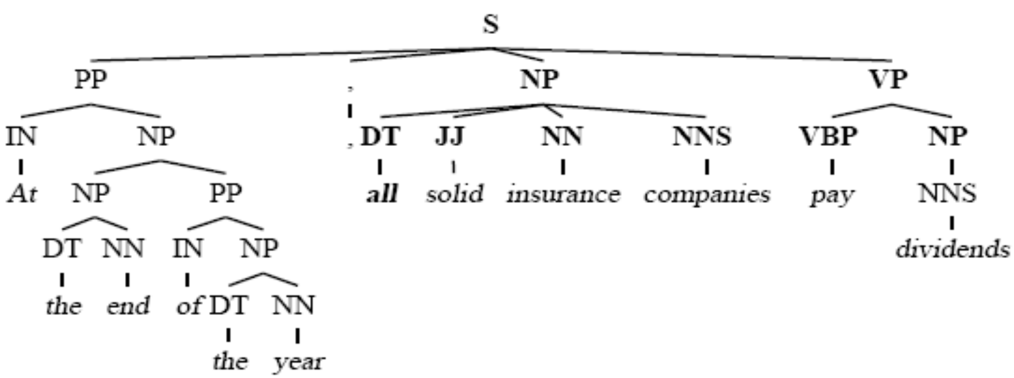
# Our Model: an example

***Intra-pair operations***

$T_1$



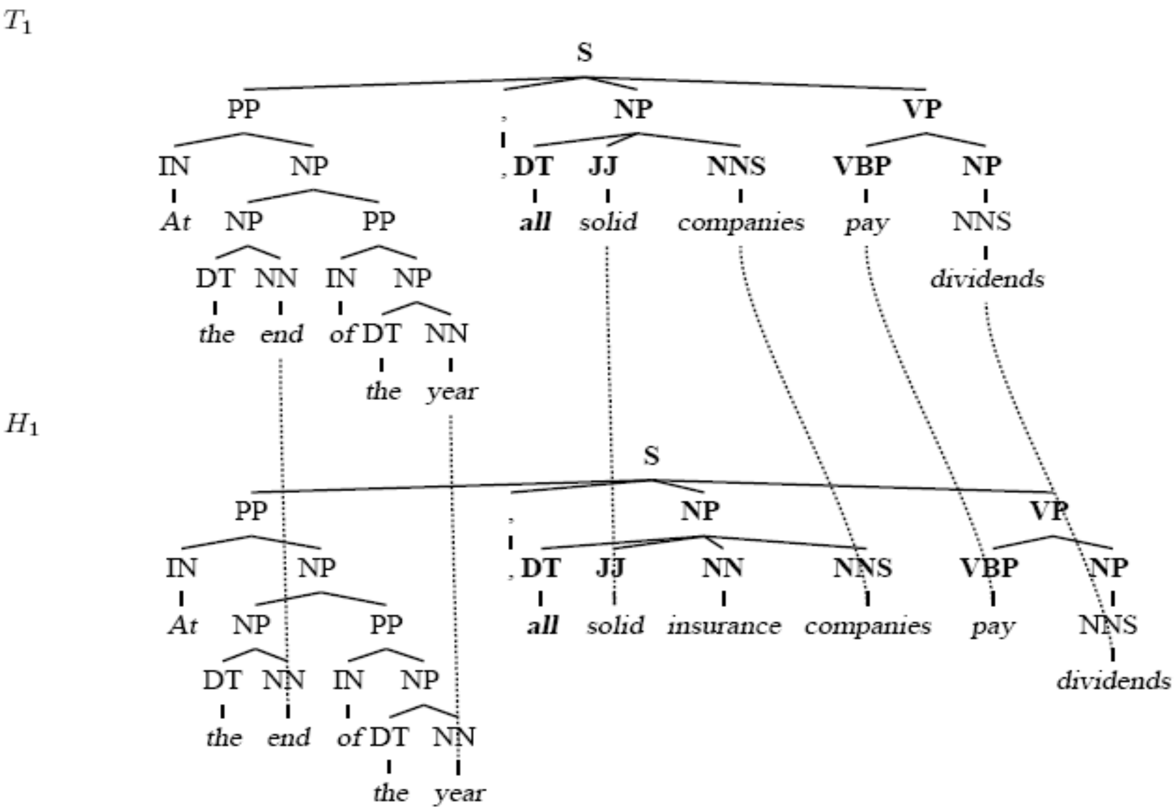

$H_1$



# Our Model: an example

***Intra-pair operations***

→ Finding ***anchors***

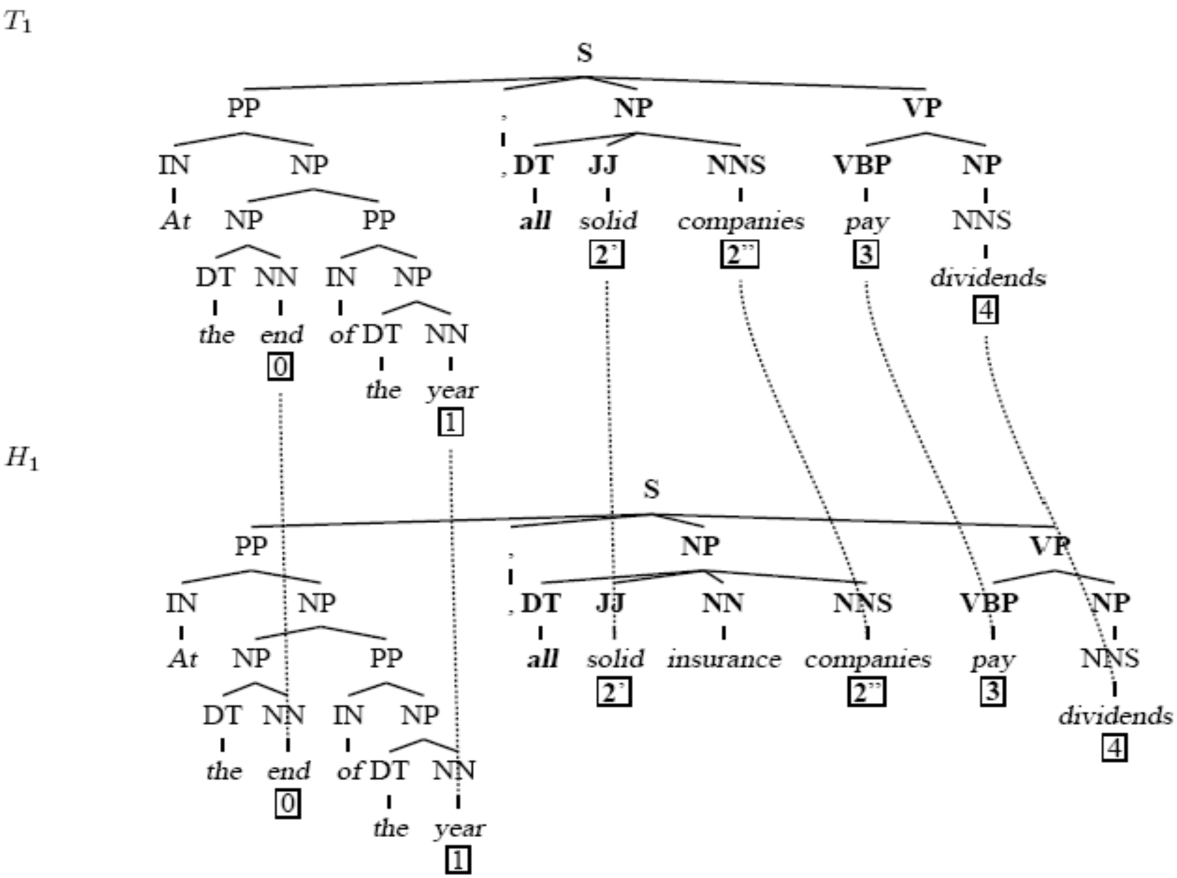


# Our Model: an example

## ***Intra-pair operations***

→Finding ***anchors***

→Naming anchors with ***placeholders***



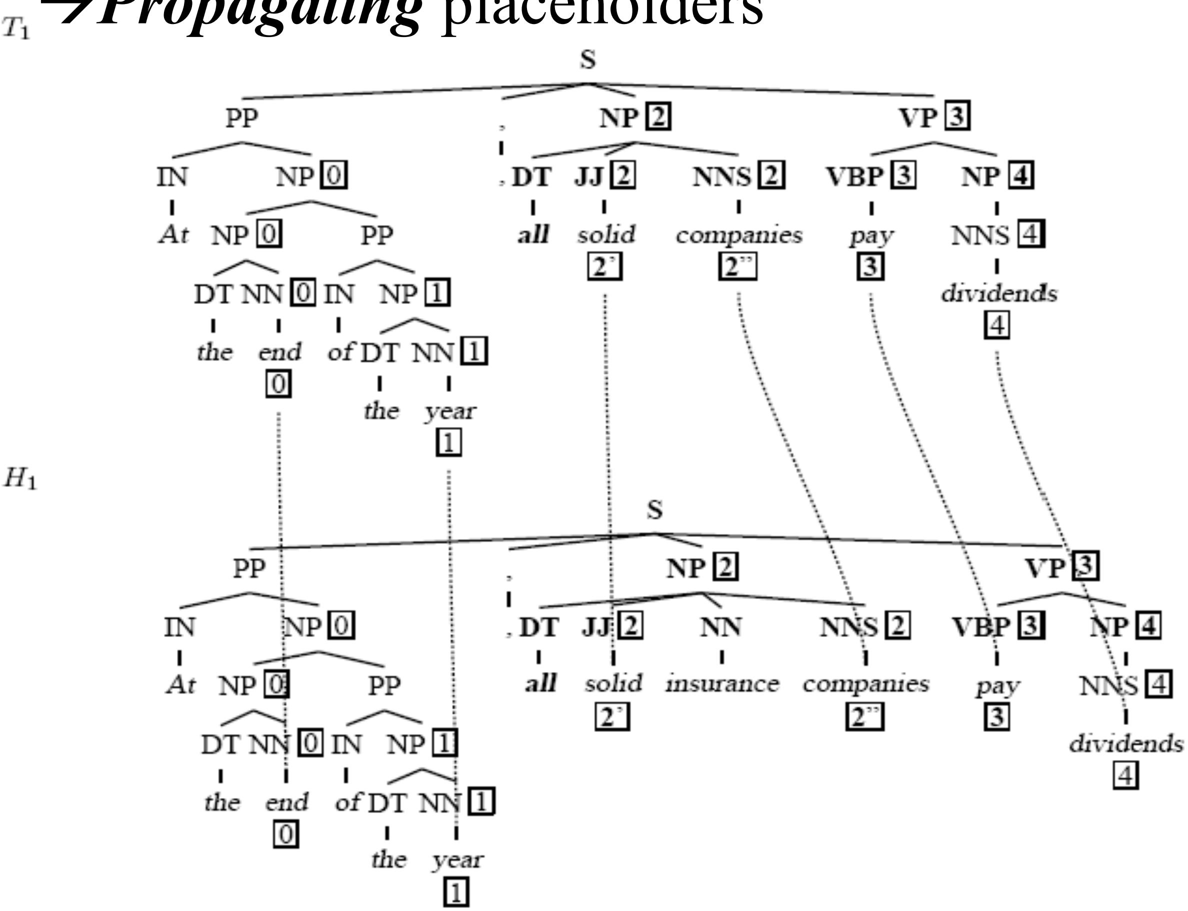
# Our Model: an example

## *Intra-pair operations*

→Finding ***anchors***

→Naming anchors with ***placeholders***

→***Propagating*** placeholders



# Our Model: an example

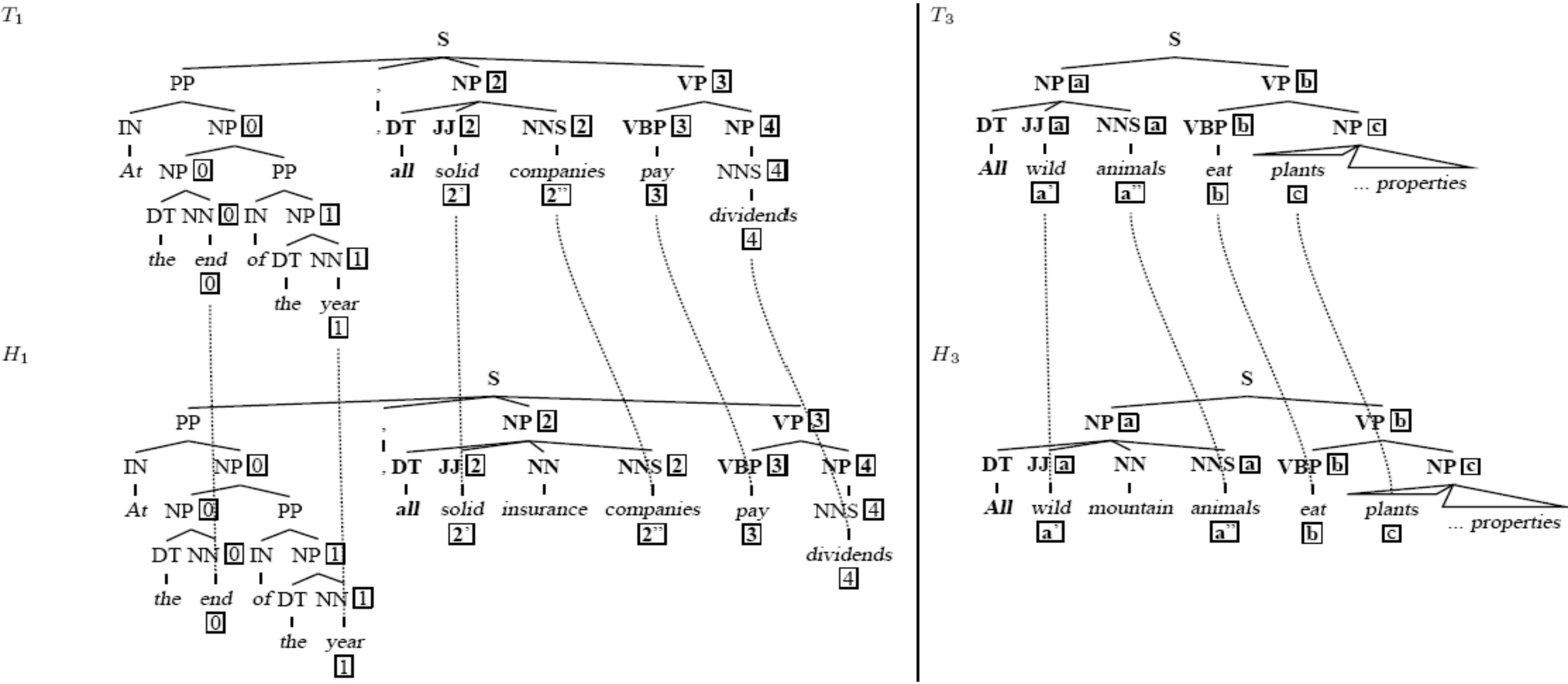
## *Intra-pair operations*

## *Cross-pair operations*

→Finding ***anchors***

→Naming anchors with ***placeholders***

→***Propagating*** placeholders



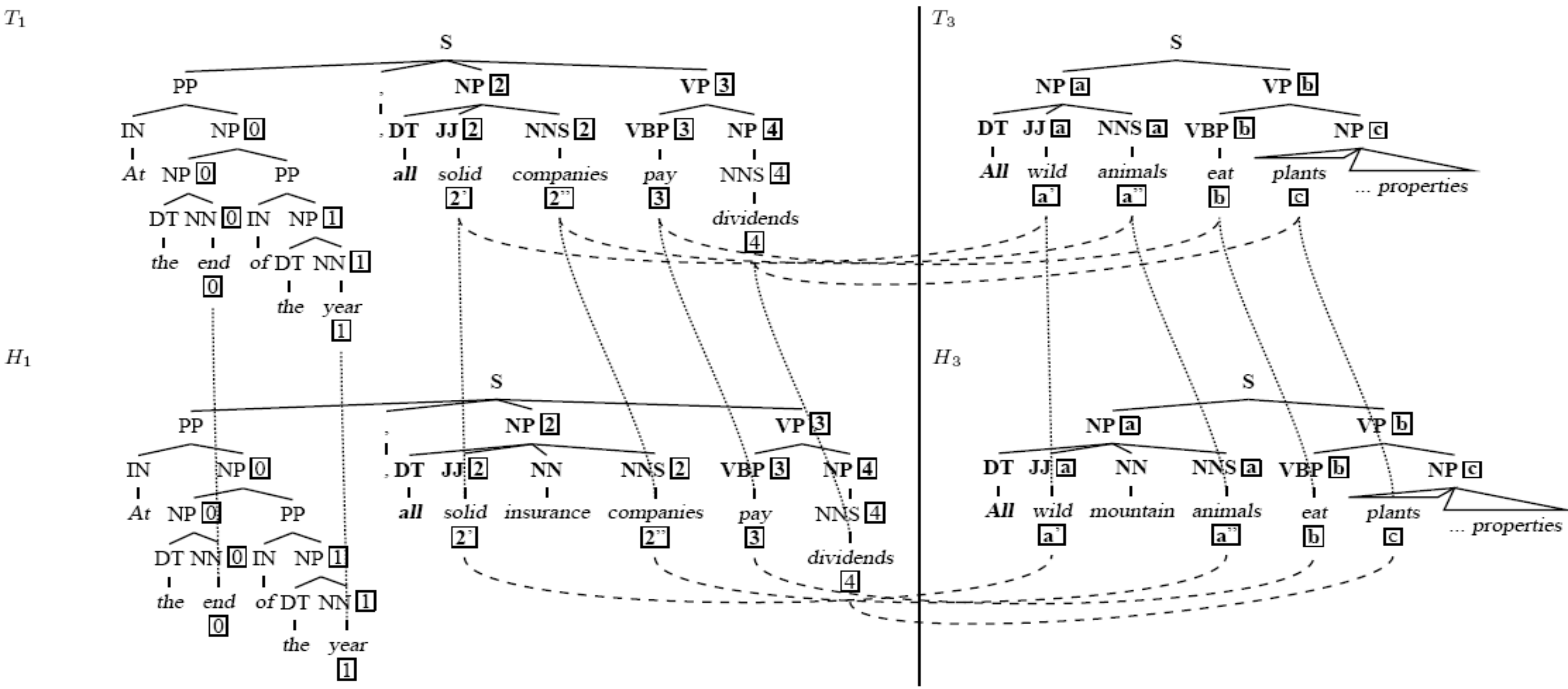
# Our Model: an example

## *Intra-pair operations*

→Finding ***anchors***
→Naming anchors with ***placeholders***
→***Propagating*** placeholders

## *Cross-pair operations*

→Matching placeholders across pairs



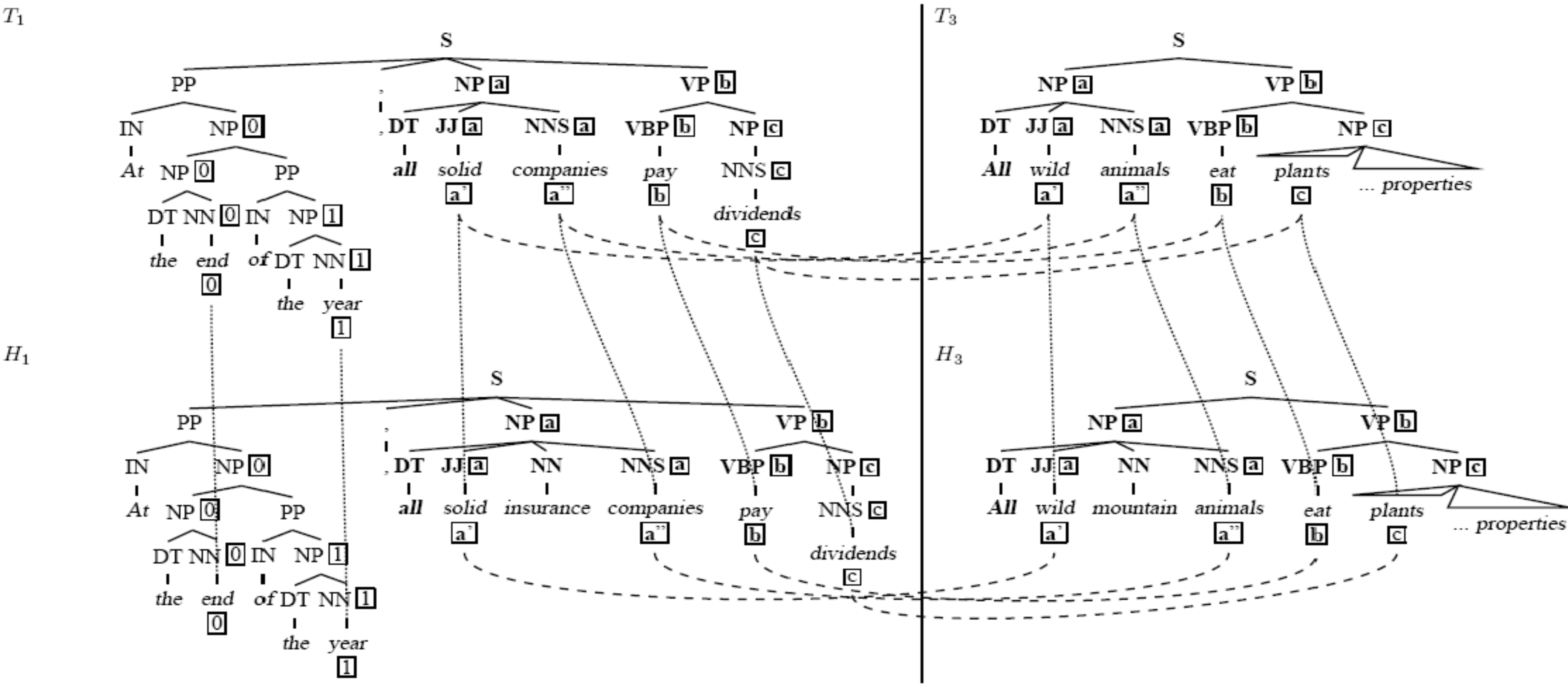
# Our Model: an example

## *Intra-pair operations*

→Finding ***anchors***
→Naming anchors with ***placeholders***
→***Propagating*** placeholders

## *Cross-pair operations*

→Matching placeholders across pairs
→Renaming placeholders



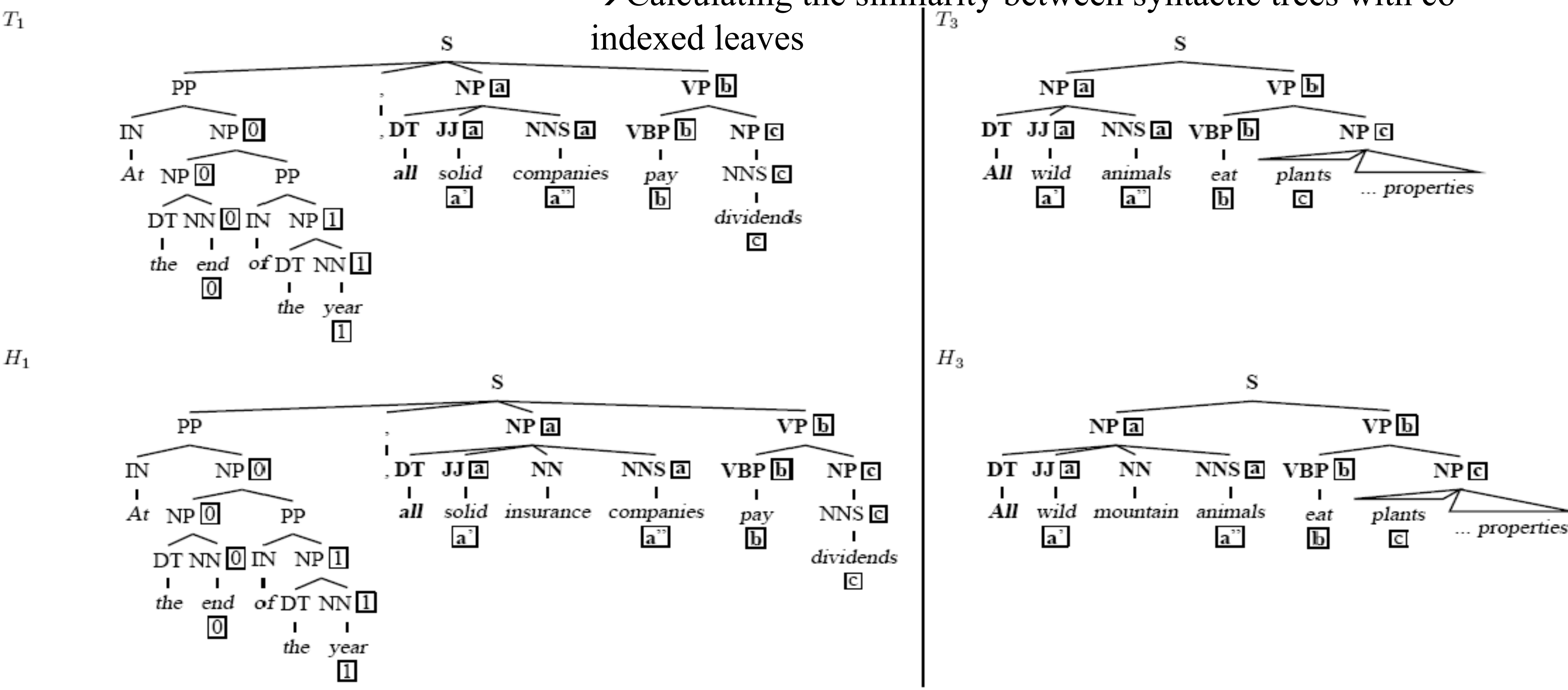
# Our Model: an example

## *Intra-pair operations*

→Finding ***anchors***
→Naming anchors with ***placeholders***
→***Propagating*** placeholders

## *Cross-pair operations*

→Matching placeholders across pairs
→Renaming placeholders
→Calculating the similarity between syntactic trees with co-indexed leaves



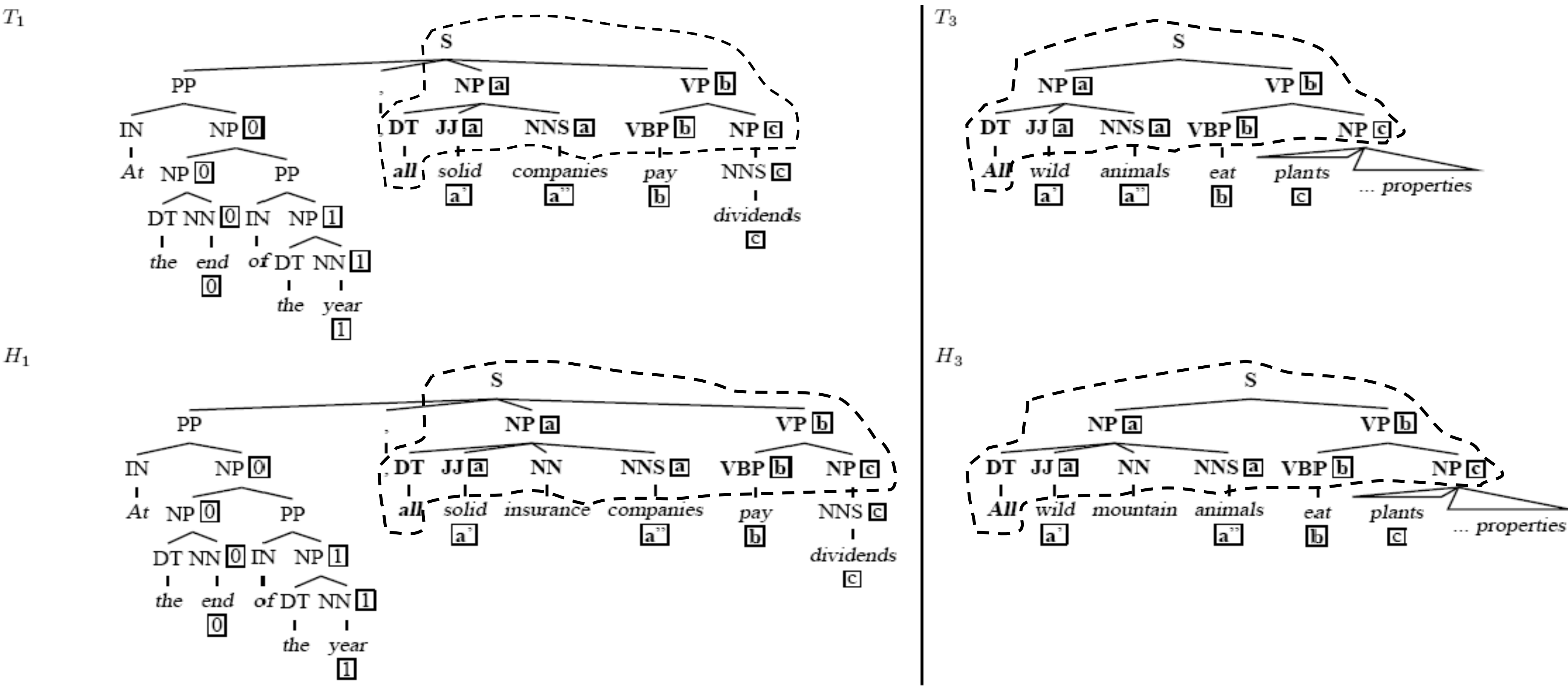
# Our Model: an example

## *Intra-pair operations*

→Finding ***anchors***
→Naming anchors with ***placeholders***
→***Propagating*** placeholders

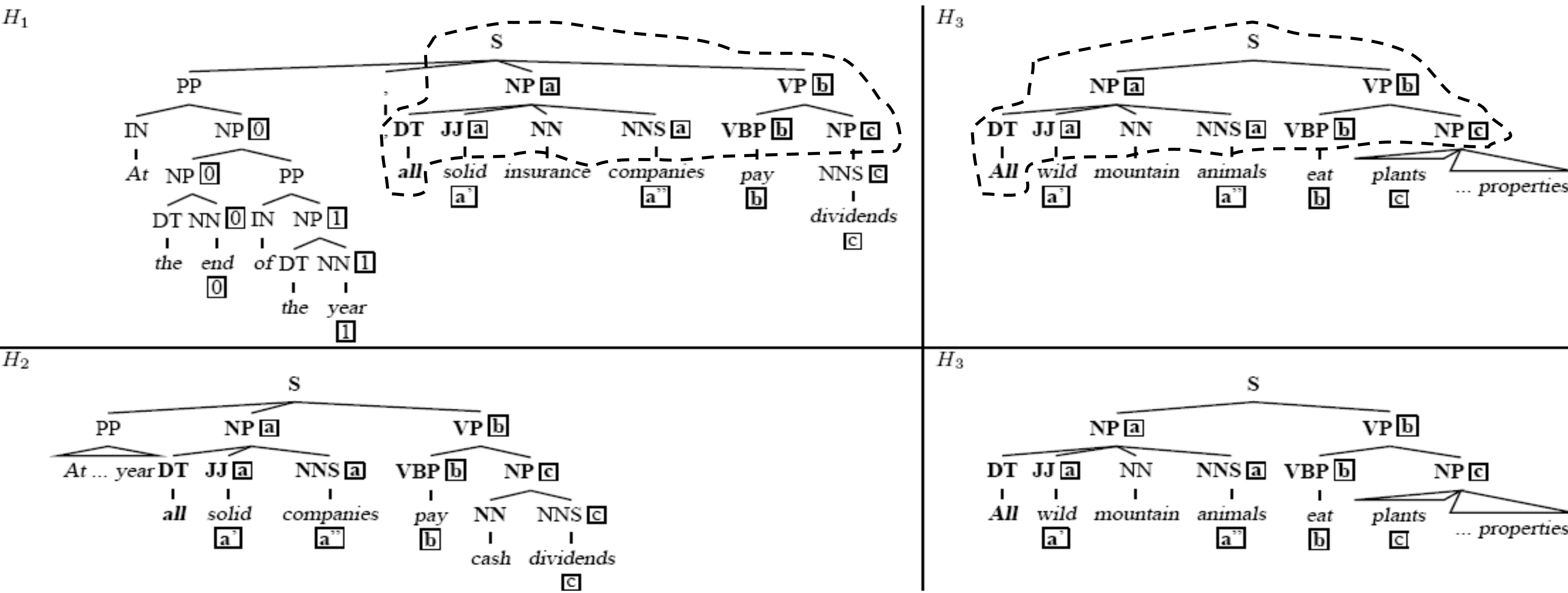
## *Cross-pair operations*

→Matching placeholders across pairs
→Renaming placeholders
→Calculating the similarity between syntactic trees with co-indexed leaves



# Our Model: an example

- The initial example: sim(H1,H3) > sim(H2,H3)?



# The final kernel

$$K_s((T',H'),(T'',H'')) =$$

$$\max_{c \in C}\left(K_T(t(H',c),t(H'',i)) + K_T(t(T',c),t(T'',i))\right)$$

where:

- ***c*** is an assignment of placeholders
- ***t*** transforms the trees according to the assigned placeholders

# Experimental Results

- RTE1 (1st Recognising Textual Entailment Challenge) [Dagan et al., 2005]
  - 567 training and 800 test examples
- RTE2, [Bar Haim et al., 2006]
  - 800 training and 800 test examples

| | BOW+LS | + *TK* | + $K_{ent}$ | System Avg. |
|---|---|---|---|---|
| RTE1 | 0.5888 | 0.6213 | 0.6300 | 0.54 |
| RTE2 | 0.6038 | 0.6238 | 0.6388 | 0.59 |

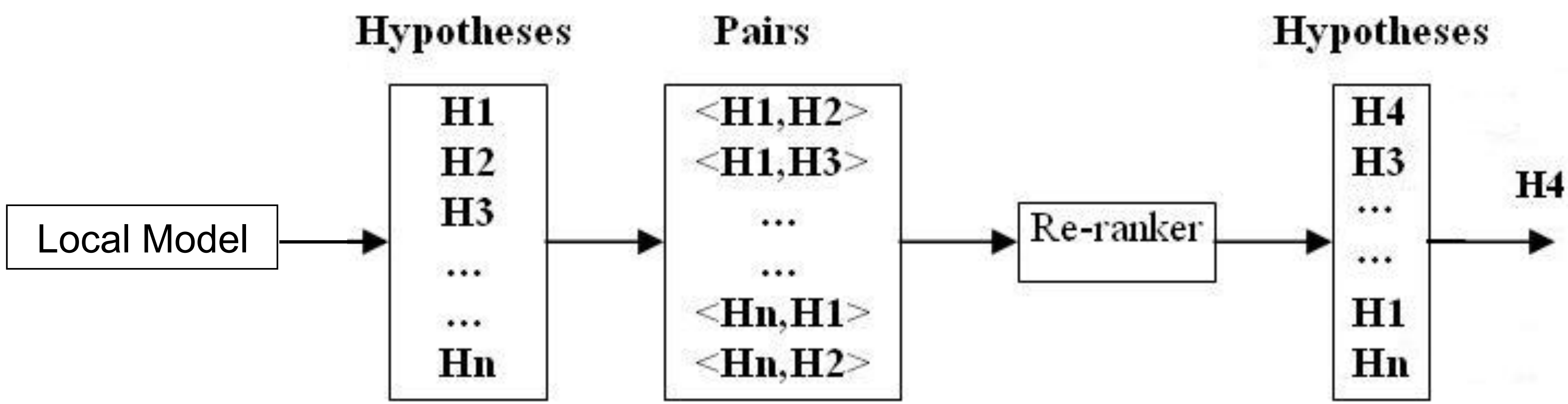
| System | Strategy | Decision | An. Level | Knowledge Resources | Acc. |
|---|---|---|---|---|---|
| (Hickl et al., 2006) | lex,syn,trg | mlr | lxs,synt | WN,paraph,PropBank | 0.7538 |
| (Tatu and Moldovan, 2006) | lex | thr,inf | sur,sem | WN,SUMO,ExtWN,axioms | 0.7375 |
| (Zanzotto et al., 2006) | syn | mlr | lxs,syn | WN | 0.6388 |
| (Adams, 2006) | lex | mlr | sur,lxs | WN | 0.6262 |
| (Bos and Markert, 2006) | lex | mlr,inf | sur,lxs | WN,axioms | 0.6160 |
| (Kouylekov and Magnini, 2006) | synt | thr,mlr | lxs,syn | WN,DIRT | 0.6050 |
| (MacCartney et al., 2006) | synt | mlr | lxs,syn | WN | 0.6050 |
| (Snow et al., 2006) | trg,lex | rul,mlr | lxs,syn | WN,MindNet, thes | 0.6025 |
| (Herrera et al., 2006) | lex,syn | mlr | lex,syn | WN | 0.5975 |
| (Nielsen et al., 2006) | lex,syn | mlr | sur,syn | | 0.5960 |
| (Marsi et al., 2006) | syn | thr | lxs,syn | WN | 0.5960 |
| (Katrenko and Adriaans, 2006) | lex,syn | mlr | syn | | 0.5900 |
| (Burchardt and Frank, 2006) | syn | mlr | lxs,syn | WN,FrameNet,SUMO | 0.5900 |
| (Rus, 2006) | syn | thr | lxs,syn | WN | 0.5900 |
| (Litkowski, 2006) | lex | thr | sur | | 0.5810 |
| (Inkpen et al., 2006) | trg,lex | mlr | lxs,syn | WN | 0.5800 |
| (Ferrndez et al., 2006) | syn | thr | lex,syn | WN | 0.5563 |
| (Schilder and McInnes, 2006) | lex,syn | mlr | lxs,syn | WN | 0.5550 |

# KERNELS FOR RE-RANKING

# Re-ranking framework

- Local classifier generates the most likely set of hypotheses.
- These are used to build annotation pairs, $\langle H^i, H^j \rangle$.
  - positive instances if $H^i$ *more correct* than $H^j$,
- A binary classifier decides if $H^i$ is more accurate than $H^j$.
- Each candidate annotation $H^i$ is described by a structural representation
- This way Kernels can exploit all dependencies between **features and labels**

# Re-ranking framework



# Syntactic Parsing Re-ranking

- Pairs of parse trees (Collins and Duffy, 2002)
- N-best parse generated by the Collins' parser
- Re-ranking using STK in a perceptron algorithm

# SPOKEN LANGUAGE UNDERSTANDING

# Concept Segmentation and Classification task

- Given a transcription, i.e. a sequence of words, chunk and label subsequences with concepts
- Air Travel Information System (ATIS)
  - Dialog systems answering user questions
  - Conceptually annotated dataset
  - Frames

# An example of concept annotation in ATIS

- User request: ***list TWA flights from Boston to Philadelphia***

$$\underbrace{list}_{null}\ \underbrace{TWA}_{airline\_code}\ \underbrace{flights}_{null}\ \underbrace{from}_{null}\ \underbrace{Boston}_{fromloc.city}\ \underbrace{to}_{null}\ \underbrace{Philadelphia}_{toloc.city}$$

- The concepts are used to build rules for the dialog manager (e.g. actions for using the DB)
  - from location
  - to location
  - airline code

| | |
|---|---|
| list flights from boston to Philadelphia | |
| FRAME: | FLIGHT |
| | FROMLOC.CITY = boston |
| | TOLOC.CITY = Philadelphia |

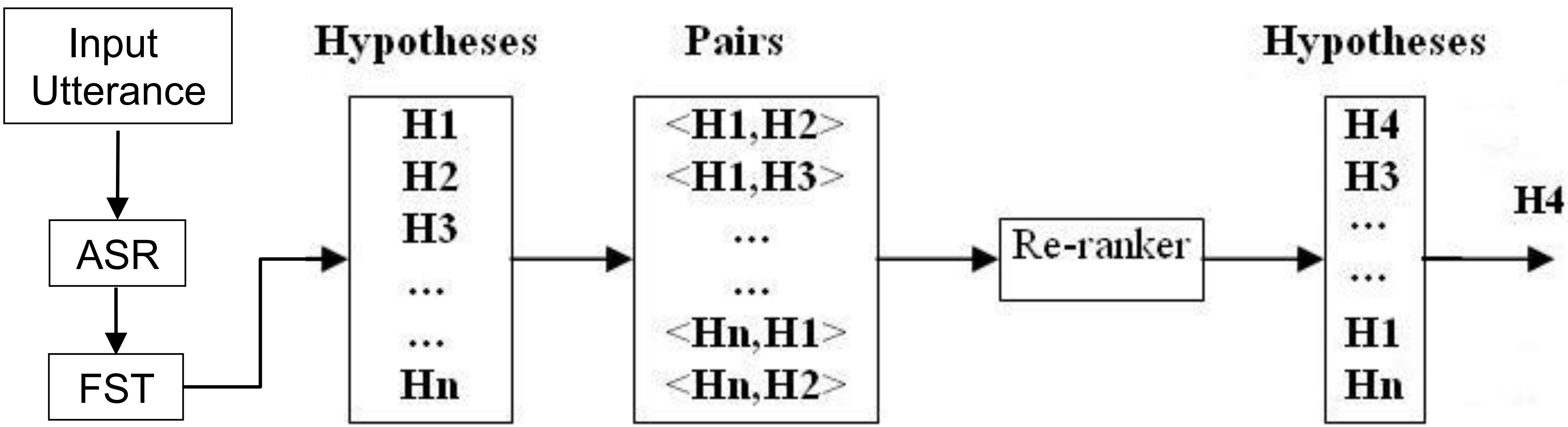
# Our Approach
## [Dinarelli et al., SLT 2008-10, Interspeech 2009]

- Use of Finite State Transducer (or CRF) to generate word sequences and concepts
- Probability of each annotation

$\Rightarrow$ *m* best hypothesis can be generated

- Idea: use a discriminative model to choose the best one
  - Re-ranking and selecting the top one

# Re-ranking for SLU



# Re-ranking concept labeling

- *I have a problem with my monitor*

***H***$^i$: *I* ***NULL*** *have* ***NULL*** *a* ***PROBLEM-B*** *problem* ***PROBLEM-I*** *with* ***NULL*** *my* ***HW-B*** *monitor* ***HW-I***

***H***$^j$: *I* ***NULL*** *have* ***NULL*** *a* ***NULL*** *problem* ***HW-B*** *with* ***NULL*** *my* ***NULL*** *monitor*

# Luna Corpus

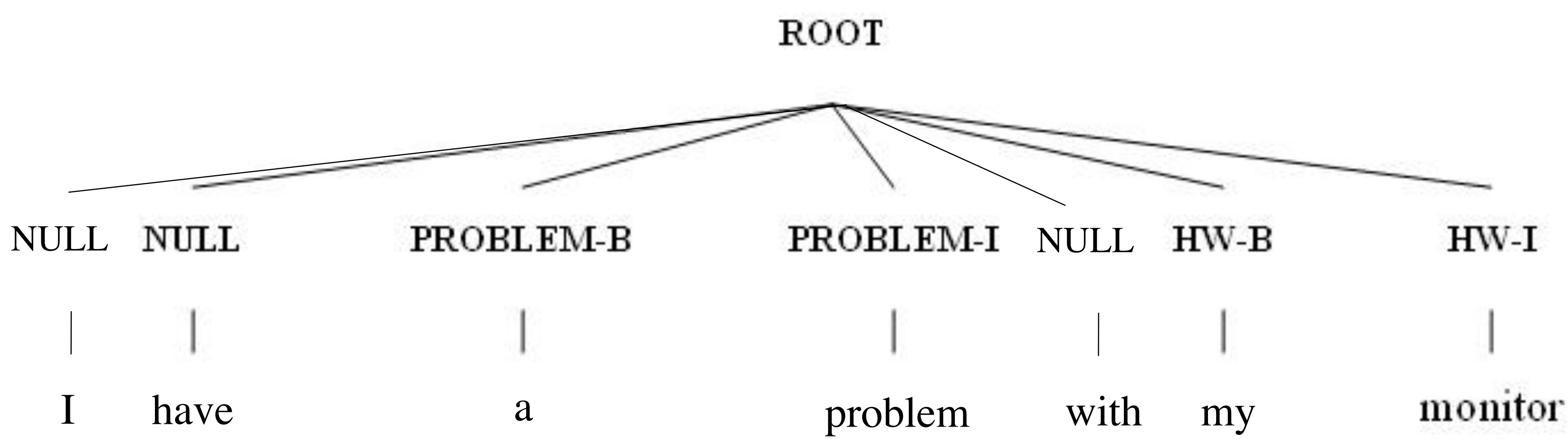
- Wizard of OZ, helpdesk scenario

| Corpus LUNA | Training set | | Test set | |
|---|---|---|---|---|
| | words | concepts | words | concepts |
| Dialogs | 183 | | 67 | |
| Turns | 1,019 | | 373 | |
| Tokens | 8,512 | 2,887 | 2,888 | 984 |
| Vocabulary | 1,172 | 34 | - | - |
| OOV rate | - | - | 3.2% | 0.1% |

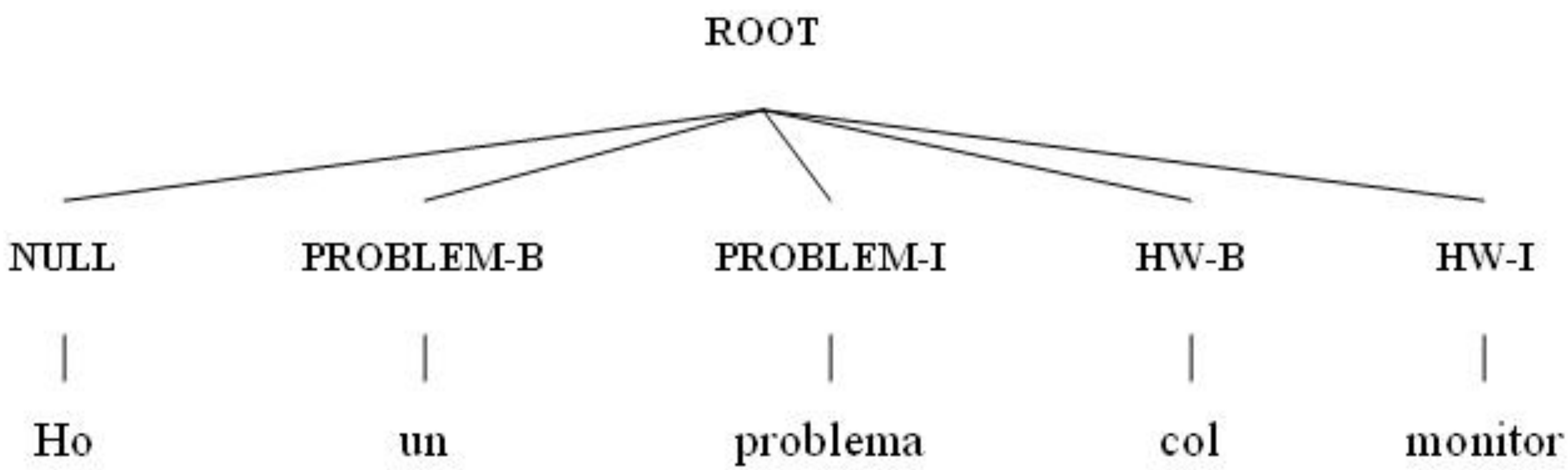
# Media Corpus

| | training | | development | | test | |
|---|---|---|---|---|---|---|
| # sentences | 12,908 | | 1,259 | | 3,005 | |
| | words | concepts | words | concepts | words | concepts |
| # tokens | 94,466 | 43,078 | 10,849 | 4,705 | 25,606 | 11,383 |
| # vocabulary | 2,210 | 99 | 838 | 66 | 1,276 | 78 |
| # OOV rate [%] | – | – | 1.33 | 0.02 | 1.39 | 0.04 |

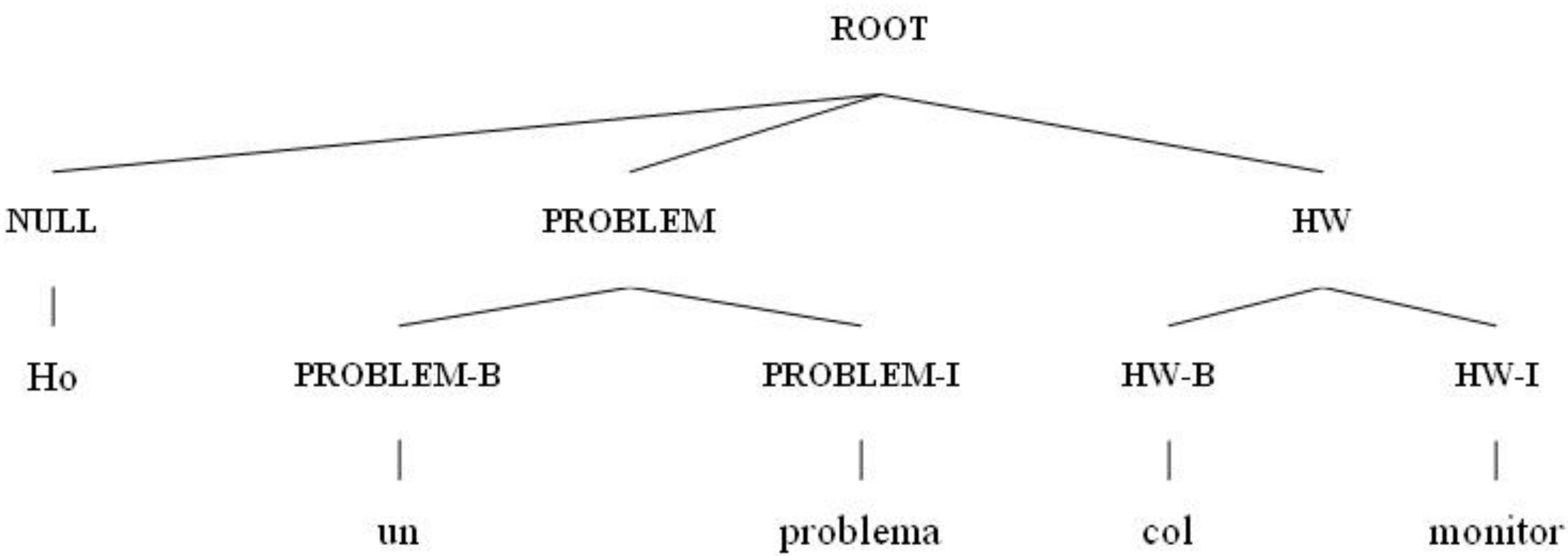
# Flat tree representation



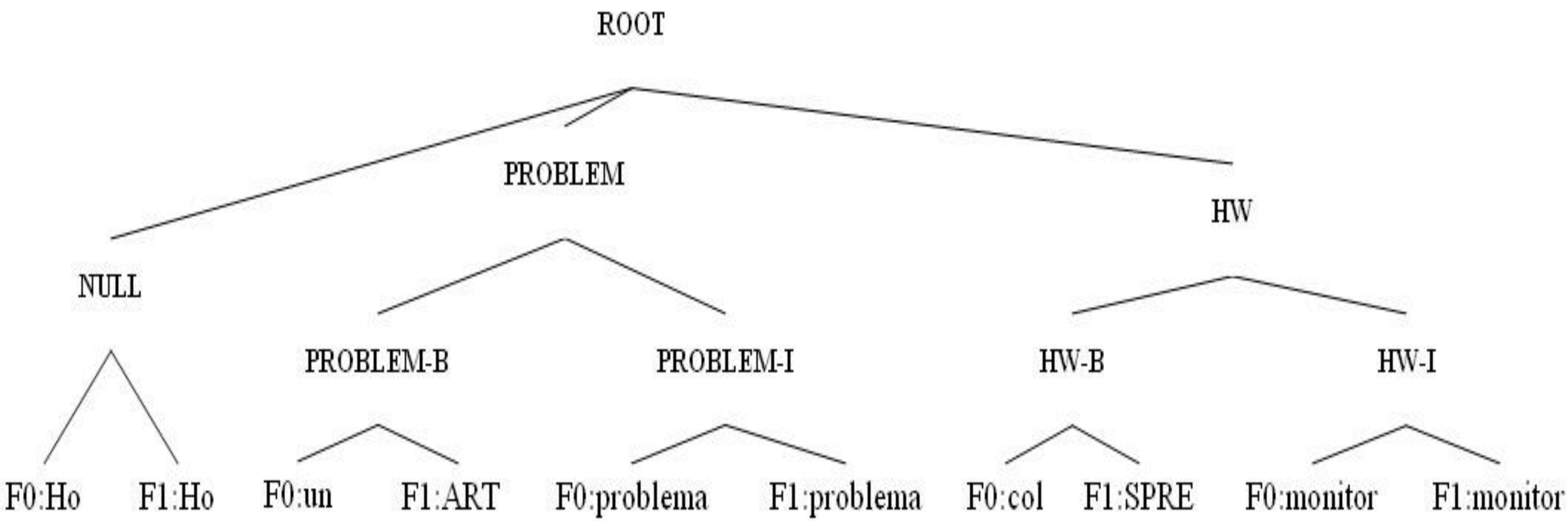
# Cross-language approach: Italian version



# Multilevel Tree



# Enriched Multilevel Tree



# Results on LUNA

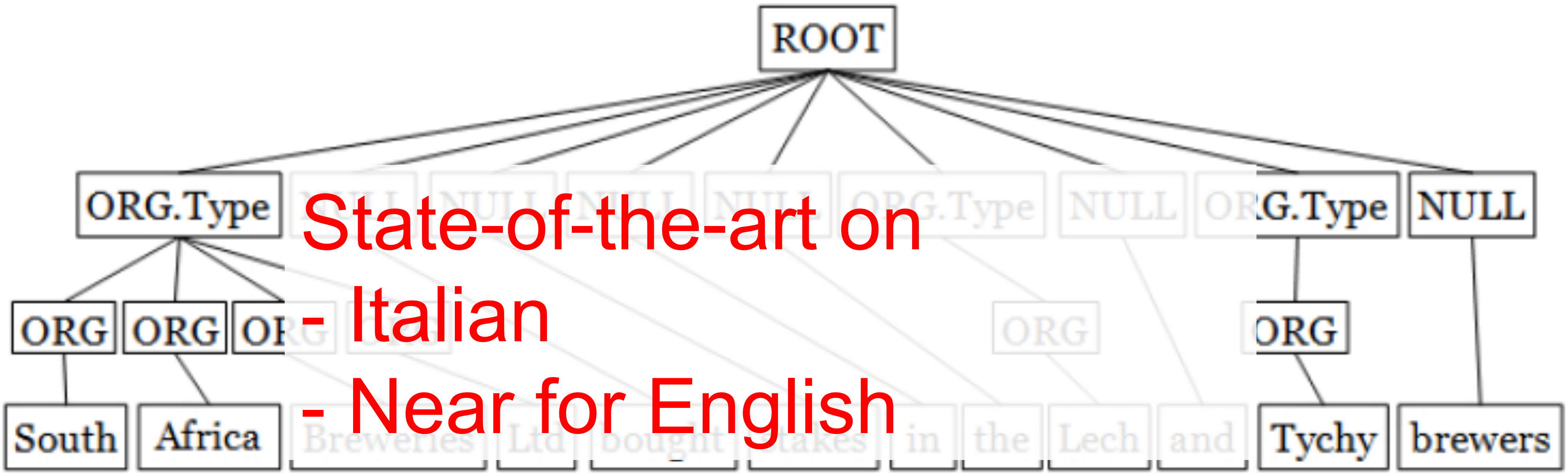
| Model | Text Input (CER) | | Speech Input (CER) | |
|---|---|---|---|---|
| | Attr. | Attr.-Val. | Attr. | Attr.-Val. |
| **FST** | 24.4% | 27.4% | 36.4% | 39.9% |
| **SVM** | 25.3% | 27.1% | 34.0% | 36.7% |
| **CRF** | 21.3% | 23.5% | 31.0% | 34.2% |
| **FST-RR** | 20.7% | 22.8% | 32.7% | 36.2% |
| **CRF-RR** | 19.9% | 21.9% | 29.0% | 32.2% |
| $FST + RR_S$ | 19.2% | 21.5% | 30.4% | 33.8% |
| $CRF + RR_S$ | 19.0% | 21.1% | 28.3% | 31.4% |

# Results on Media

| Model | Text Input (CER) Attr. | Text Input (CER) Attr.-Val. | Speech Input (CER) Attr. | Speech Input (CER) Attr.-Val. |
|---|---|---|---|---|
| FST | 14.2% | 17.0% | 28.9% | 33.6% |
| SVM | [illegible] | 15.0% | 25.8% | 29.7% |
| CRF | 11.7% | 14.2% | 24.3% | 28.2% |
| FST-RR | 11.9% | 14.6% | 25.4% | 29.9% |
| CRF-RR | 11.5% | 14.1% | 23.6% | 27.2% |
| $FST + RR_S$ | 11.3% | 13.8% | 24.5% | 28.2% |
| $CRF + RR_S$ | 11.1% | 13.1% | 22.7% | 26.3% |

State-of-the-art on
- Luna
- Media

# Re-ranking for Named-Entity Recognition
## [Vien et al, 2010]

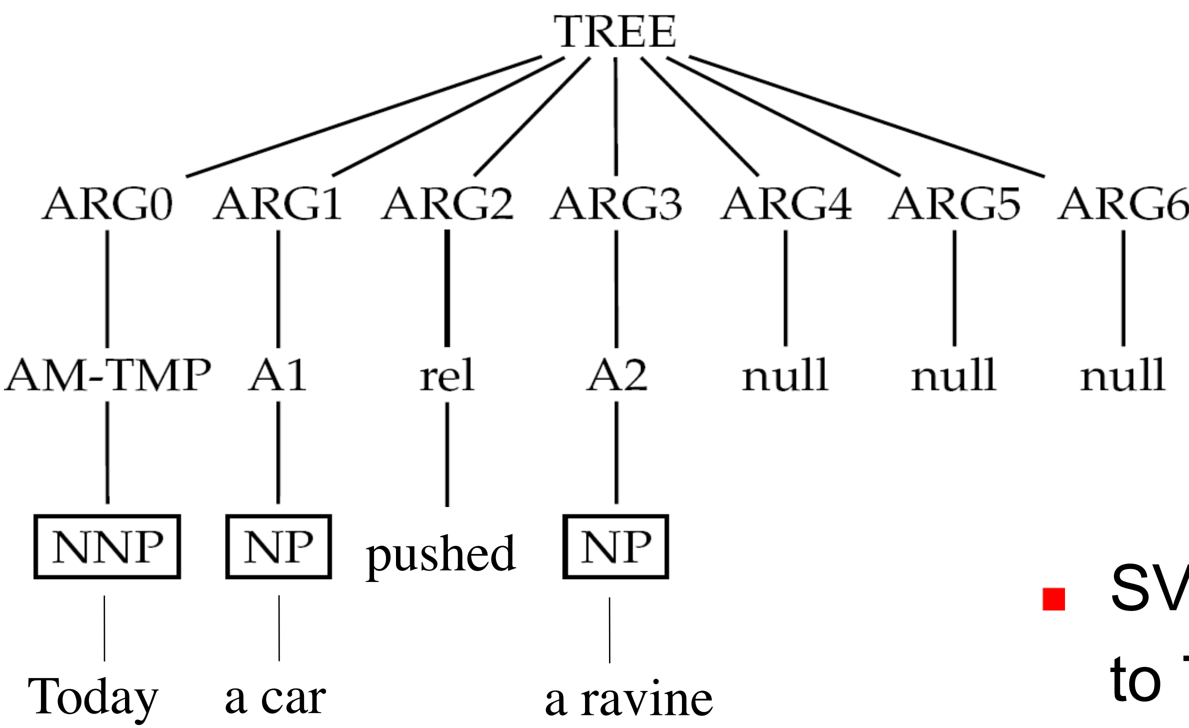



State-of-the-art on
- Italian
- Near for English

- CRF F1 from 84.86 to 88.16
- Best Italian system F1 82, improved to 84.33

# Re-ranking Predicate Argument Structures
## [Moschitti et al, CoNLL 2006]

- Today, a car was pushed into a ravine.




- SVMs F1 from 75.89 to 77.25

# Conclusions

- We used powerful ML algorithms
  - e.g. Support Vector Machines
  - robust to noise
- Abstract representations of examples
  - Similarity functions (Kernel Methods)
  - Structural syntactic/semantic similarity
- Modeling Question/Answer with: advanced syntactic and shallow semantic structures and relational marker
- Experiments demonstrate the benefit of such approach on
  - TREC
  - The Grand Jeopardy! Challenge (good impact on Watson)

# Conclusions (cont'd)

- Kernel methods and SVMs are useful tools to design language applications
- Basic general kernel functions can be used to engineer new kernels
- Little effort in selecting and marking/tailoring/decorating/ designing trees or designing sequences
- Easy modeling produces state-of-the-art accuracy in many tasks, SRL, RE, CR, QA, NER, SLU, RTE
- Fast prototyping and model adaptation

# Future (on going work)

- Modeling more than one sentence with deeper structures: shallow semantics and *discourse*
- The objective is more compact and accurate models applicable to whole paragraphs.
- Use of reverse kernel engineering to study linguistic phenomena:
  - [Pighin&Moschitti, CoNLL2009, EMNLP2009, CoNLL2010]
  - To mine the most relevant fragments according to SVMs gradient
  - To use the linear space

Thank you

# References


- Alessandro Moschitti’ handouts http://disi.unitn.eu/~moschitt/teaching.html
- Alessandro Moschitti and Silvia Quarteroni, *Linguistic Kernels for Answer Re-ranking in Question Answering Systems,* Information and Processing Management, ELSEVIER, 2010*.*
- Yashar Mehdad, Alessandro Moschitti and Fabio Massimo Zanzotto. *Syntactic/ Semantic Structures for Textual Entailment Recognition*. Human Language Technology - North American chapter of the Association for Computational Linguistics (HLT-NAACL), 2010, Los Angeles, Calfornia.
- Daniele Pighin and Alessandro Moschitti. *On Reverse Feature Engineering of Syntactic Tree Kernels*. In Proceedings of the 2010 Conference on Natural Language Learning, Upsala, Sweden, July 2010. Association for Computational Linguistics.
- Thi Truc Vien Nguyen, Alessandro Moschitti and Giuseppe Riccardi. *Kernel-based Reranking for Entity Extraction.* In proceedings of the 23rd International Conference on Computational Linguistics (COLING), August 2010, Beijing, China.

# References


- Alessandro Moschitti. *Syntactic and semantic kernels for short text pair categorization*. In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pages 576–584, Athens, Greece, March 2009.
- Truc-Vien Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. *Convolution kernels on constituent, dependency and sequential structures for relation extraction*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1378–1387, Singapore, August 2009.
- Marco Dinarelli, Alessandro Moschitti, and Giuseppe Riccardi. *Re-ranking models based-on small training data for spoken language understanding*. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1076–1085, Singapore, August 2009.
- Alessandra Giordani and Alessandro Moschitti. *Syntactic Structural Kernels for Natural Language Interfaces to Databases*. In ECML/PKDD, pages 391–406, Bled, Slovenia, 2009.

# References


- Alessandro Moschitti, Daniele Pighin and Roberto Basili*. Tree Kernels for Semantic Role Labeling,* Special Issue on Semantic Role Labeling, Computational Linguistics Journal. March 2008.
- Fabio Massimo Zanzotto, Marco Pennacchiotti and Alessandro Moschitti*, A Machine Learning Approach to Textual Entailment Recognition,* Special Issue on Textual Entailment Recognition, Natural Language Engineering, Cambridge University Press., 2008
- Mona Diab, Alessandro Moschitti, Daniele Pighin, *Semantic Role Labeling Systems for Arabic Language using Kernel Methods*. In proceedings of the 46th Conference of the Association for Computational Linguistics (ACL'08). Main Paper Section. Columbus, OH, USA, June 2008.
- Alessandro Moschitti, Silvia Quarteroni, Kernels on Linguistic Structures for Answer Extraction. In proceedings of the 46th Conference of the Association for Computational Linguistics (ACL'08). Short Paper Section. Columbus, OH, USA, June 2008.

# References


- Yannick Versley, Simone Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang and Alessandro Moschitti, *BART: A Modular Toolkit for Coreference Resolution*, In Proceedings of the Conference on Language Resources and Evaluation, Marrakech, Marocco, 2008.
- Alessandro Moschitti, *Kernel Methods, Syntax and Semantics for Relational Text Categorization*. In proceeding of ACM 17th Conference on Information and Knowledge Management (CIKM). Napa Valley, California, 2008.
- Bonaventura Coppola, Alessandro Moschitti, and Giuseppe Riccardi. *Shallow semantic parsing for spoken language understanding*. In Proceedings of HLT-NAACL Short Papers, pages 85–88, Boulder, Colorado, June 2009. Association for Computational Linguistics.
- Alessandro Moschitti and Fabio Massimo Zanzotto, *Fast and Effective Kernels for Relational Learning from Texts*, Proceedings of The 24th Annual International Conference on Machine Learning (ICML 2007).

# References


- Alessandro Moschitti, Silvia Quarteroni, Roberto Basili and Suresh Manandhar, *Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classification*, Proceedings of the 45th Conference of the Association for Computational Linguistics (ACL), Prague, June 2007.
- Alessandro Moschitti and Fabio Massimo Zanzotto, *Fast and Effective Kernels for Relational Learning from Texts*, Proceedings of The 24th Annual International Conference on Machine Learning (ICML 2007), Corvallis, OR, USA.
- Daniele Pighin, Alessandro Moschitti and Roberto Basili, *RTV: Tree Kernels for Thematic Role Classification*, Proceedings of the 4th International Workshop on Semantic Evaluation (SemEval-4), English Semantic Labeling, Prague, June 2007.
- Stephan Bloehdorn and Alessandro Moschitti, *Combined Syntactic and Semanitc Kernels for Text Classification*, to appear in the 29th European Conference on Information Retrieval (ECIR), April 2007, Rome, Italy.
- Fabio Aiolli, Giovanni Da San Martino, Alessandro Sperduti, and Alessandro Moschitti, *Efficient Kernel-based Learning for Trees*, to appear in the IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Honolulu, Hawaii, 2007

# References


- Alessandro Moschitti, Silvia Quarteroni, Roberto Basili and Suresh Manandhar, *Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classification*, Proceedings of the 45th Conference of the Association for Computational Linguistics (ACL), Prague, June 2007.
- Alessandro Moschitti, Giuseppe Riccardi, Christian Raymond, *Spoken Language Understanding with Kernels for Syntactic/Semantic Structures*, Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU2007), Kyoto, Japan, December 2007
- Stephan Bloehdorn and Alessandro Moschitti, *Combined Syntactic and Semantic Kernels for Text Classification*, to appear in the 29th European Conference on Information Retrieval (ECIR), April 2007, Rome, Italy.
- Stephan Bloehdorn, Alessandro Moschitti: Structure and semantics for expressive text kernels. In proceeding of ACM 16th Conference on Information and Knowledge Management (CIKM-short paper) 2007: 861-864, Portugal.

# References


- Fabio Aiolli, Giovanni Da San Martino, Alessandro Sperduti, and Alessandro Moschitti, *Efficient Kernel-based Learning for Trees*, to appear in the IEEE Symposium on Computational Intelligence and Data Mining (CIDM), Honolulu, Hawaii, 2007.
- Alessandro Moschitti*, Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees.* In Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany, 2006.
- Fabio Aiolli, Giovanni Da San Martino, Alessandro Sperduti, and Alessandro Moschitti, *Fast On-line Kernel Learning for Trees*, International Conference on Data Mining (ICDM) 2006 (short paper).
- Stephan Bloehdorn, Roberto Basili, Marco Cammisa, Alessandro Moschitti, *Semantic Kernels for Text Classification based on Topological Measures of Feature Similarity*. In Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 06), Hong Kong, 18-22 December 2006. (short paper).

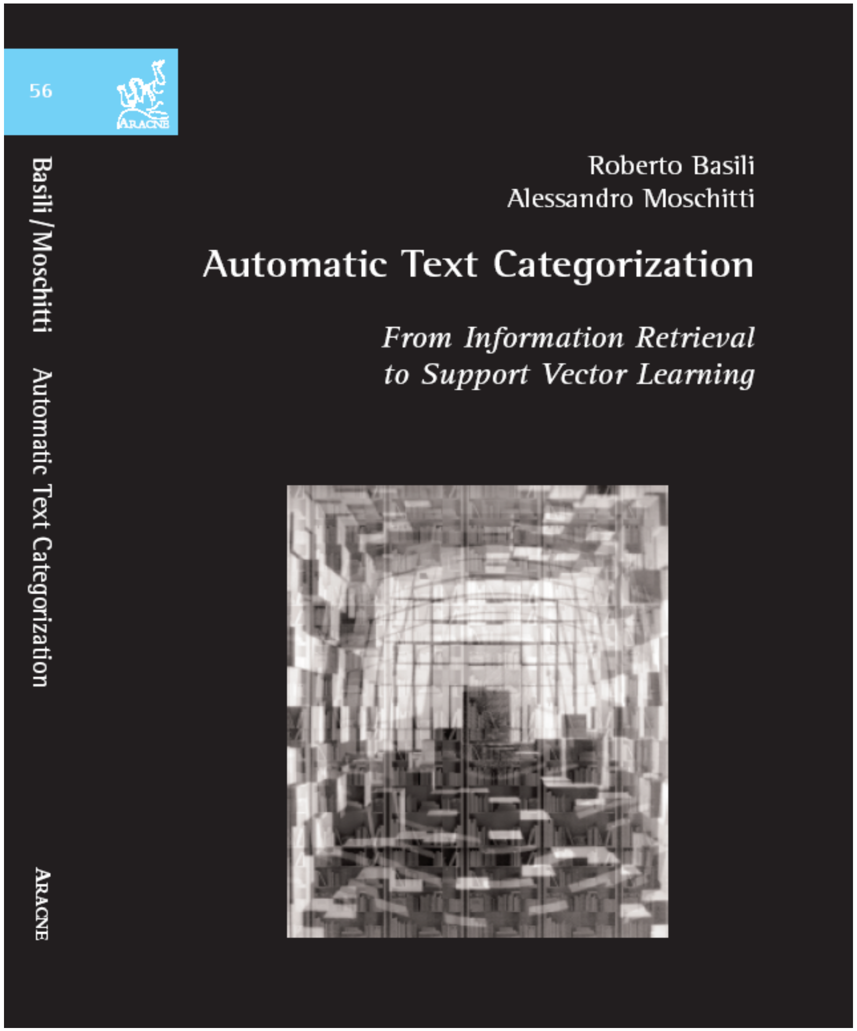
# References


- Roberto Basili, Marco Cammisa and Alessandro Moschitti, *A Semantic Kernel to classify texts with very few training examples*, in Informatica, an international journal of Computing and Informatics, 2006.
- Fabio Massimo Zanzotto and Alessandro Moschitti, *Automatic learning of textual entailments with cross-pair similarities*. In Proceedings of COLING-ACL, Sydney, Australia, 2006.
- Ana-Maria Giuglea and Alessandro Moschitti*, Semantic Role Labeling via FrameNet, VerbNet and PropBank.* In Proceedings of COLING-ACL, Sydney, Australia, 2006.
- Alessandro Moschitti*, Making tree kernels practical for natural language learning.* In Proceedings of the Eleventh International Conference on European Association for Computational Linguistics, Trento, Italy, 2006.
- Alessandro Moschitti, Daniele Pighin and Roberto Basili. *Semantic Role Labeling via Tree Kernel joint inference*. In Proceedings of the 10th Conference on Computational Natural Language Learning, New York, USA, 2006.

# References


- Roberto Basili, Marco Cammisa and Alessandro Moschitti, *Effective use of Wordnet semantics via kernel-based learning*. In Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005), Ann Arbor (MI), USA, 2005
- Alessandro Moschitti, *A study on Convolution Kernel for Shallow Semantic Parsing*. In proceedings of the 42-th Conference on Association for Computational Linguistic (ACL-2004), Barcelona, Spain, 2004.
- Alessandro Moschitti and Cosmin Adrian Bejan, *A Semantic Kernel for Predicate Argument Classification.* In proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004), Boston, MA, USA, 2004.

# An introductory book on SVMs, Kernel methods and Text Categorization



# Non-exhaustive reference list from other authors


- V. Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.
- P. Bartlett and J. Shawe-Taylor, 1998. Advances in Kernel Methods - Support Vector Learning, chapter Generalization Performance of Support Vector Machines and other Pattern Classifiers. MIT Press.
- David Haussler. 1999. Convolution kernels on discrete structures. Technical report, Dept. of Computer Science, University of California at Santa Cruz.
- Lodhi, Huma, Craig Saunders, John Shawe Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. JMLR,2000
- Schölkopf, Bernhard and Alexander J. Smola. 2001. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA.

# Non-exhaustive reference list from other authors


- N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines (and other kernel-based learning methods)* Cambridge University Press, 2002
- M. Collins and N. Duffy, New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In ACL02, 2002.
- Hisashi Kashima and Teruo Koyanagi. 2002. Kernels for semi-structured data. In Proceedings of ICML'02.
- S.V.N. Vishwanathan and A.J. Smola. Fast kernels on strings and trees. In Proceedings of NIPS, 2002.
- Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean Michel Renders. 2003. Word sequence kernels. Journal of Machine Learning Research, 3:1059–1082. D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. JMLR, 3:1083–1106, 2003.

# Non-exhaustive reference list from other authors


- Taku Kudo and Yuji Matsumoto. 2003. Fast methods for kernel-based text analysis. In Proceedings of ACL'03.
- Dell Zhang and Wee Sun Lee. 2003. Question classification using support vector machines. In Proceedings of SIGIR'03, pages 26–32.
- Libin Shen, Anoop Sarkar, and Aravind k. Joshi. Using LTAG Based Features in Parse Reranking. In Proceedings of EMNLP'03, 2003
- C. Cumby and D. Roth. Kernel Methods for Relational Learning. In Proceedings of ICML 2003, pages 107–114, Washington, DC, USA, 2003.
- J. Shawe-Taylor and N. Cristianini. Kernel Methods for Pattern Analysis. Cambridge University Press, 2004.
- A. Culotta and J. Sorensen. Dependency tree kernels for relation extraction. In Proceedings of the 42nd Annual Meeting on ACL, Barcelona, Spain, 2004.

# Non-exhaustive reference list from other authors


- Kristina Toutanova, Penka Markova, and Christopher Manning. The Leaf Path Projection View of Parse Trees: Exploring String Kernels for HPSG Parse Selection. In Proceedings of EMNLP 2004.
- Jun Suzuki and Hideki Isozaki. 2005. Sequence and Tree Kernels with Statistical Feature Mining. In Proceedings of NIPS'05.
- Taku Kudo, Jun Suzuki, and Hideki Isozaki. 2005. Boosting based parse reranking with subtree features. In Proceedings of ACL'05.
- R. C. Bunescu and R. J. Mooney. Subsequence kernels for relation extraction. In Proceedings of NIPS, 2005.
- R. C. Bunescu and R. J. Mooney. A shortest path dependency kernel for relation extraction. In Proceedings of EMNLP, pages 724–731, 2005.
- S. Zhao and R. Grishman. Extracting relations with integrated information using kernel methods. In Proceedings of the 43rd Meeting of the ACL, pages 419–426, Ann Arbor, Michigan, USA, 2005.

# Non-exhaustive reference list from other authors


- J. Kazama and K. Torisawa. Speeding up Training with Tree Kernels for Node Relation Labeling. In Proceedings of EMNLP 2005, pages 137–144, Toronto, Canada, 2005.
- M. Zhang, J. Zhang, J. Su, , and G. Zhou. A composite kernel to extract relations between entities with both flat and structured features. In Proceedings of COLING-ACL 2006, pages 825–832, 2006.
- M. Zhang, G. Zhou, and A. Aw. Exploring syntactic structured features over parse trees for relation extraction using kernel methods. Information Processing and Management, 44(2):825–832, 2006.
- G. Zhou, M. Zhang, D. Ji, and Q. Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In Proceedings of EMNLP-CoNLL 2007, pages 728–736, 2007.

# Non-exhaustive reference list from other authors


- Ivan Titov and James Henderson. Porting statistical parsers with data-defined kernels. In Proceedings of CoNLL-X, 2006
- Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring Syntactic Features for Relation Extraction using a Convolution tree kernel. In Proceedings of NAACL.
- M. Wang. A re-examination of dependency path kernels for relation extraction. In Proceedings of the 3rd International Joint Conference on Natural Language Processing-IJCNLP, 2008.