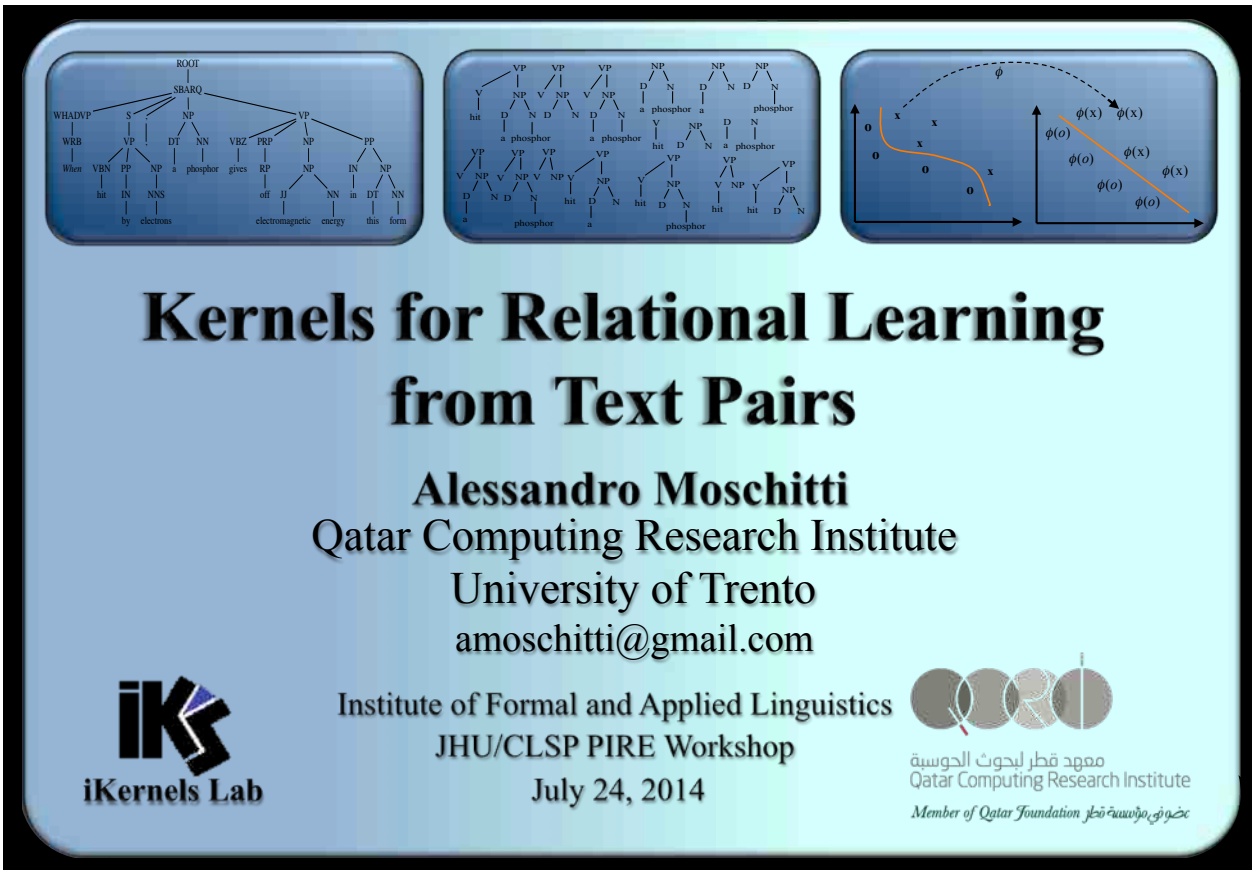


# Kernels for Relational Learning from Text Pairs

**Alessandro Moschitti**
Qatar Computing Research Institute
University of Trento
amoschitti@gmail.com

iKernels Lab

Institute of Formal and Applied Linguistics
JHU/CLSP PIRE Workshop
July 24, 2014

معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
Member of Qatar Foundation

## Outline

- Motivation
- Introduction to Structural Kernels
  - Classification function of Kernel machines
  - Kernel Definition (Kernel Trick)
  - Kernel Operators
  - String Kernel (SK), Syntactic Tree Kernel (STK), Partial Tree kernel (PTK)
  - Efficiency
- Relational kernels: Preference Reranking Kernel
- **Relational Structures** for question and answer passages
- Conclusions and future directions

## Motivation

- Linguistic relation learning regards most research in Natural Language Processing:
  - syntactic/semantic relations, coreference resolution, discourse structure, relation extraction between NEs
  - such methods typically target constituents spanning one or multiple sentences
- Relational learning from pairs of entire (short) texts
  - joint analysis of relations between different constituents
  - textual entailment, paraphrasing, correct vs. incorrect translation pairs, or question/answer pairs, etc.
- Machine learning methods are typically applied to detect such relations

## Motivation (2)

- Machine learning models use vector of features:
  - several textual similarities applied to the two texts
  - computed with different representations
- We use a different approach to relational learning from text pairs:
  - structural/linguistic representation of the text
  - semantic links between the constituents
  - structural kernels to map them in feature spaces
- Let's focus on Question Answering relations

# Let's consider: Passage Reranking

What is Mark Twain's real name?

# Passage Retrieval

**Fast Recall IR**

Roll over, Mark Twain, because Mark McGwire is on the scene.

Mark Twain couldn't have put it any better.

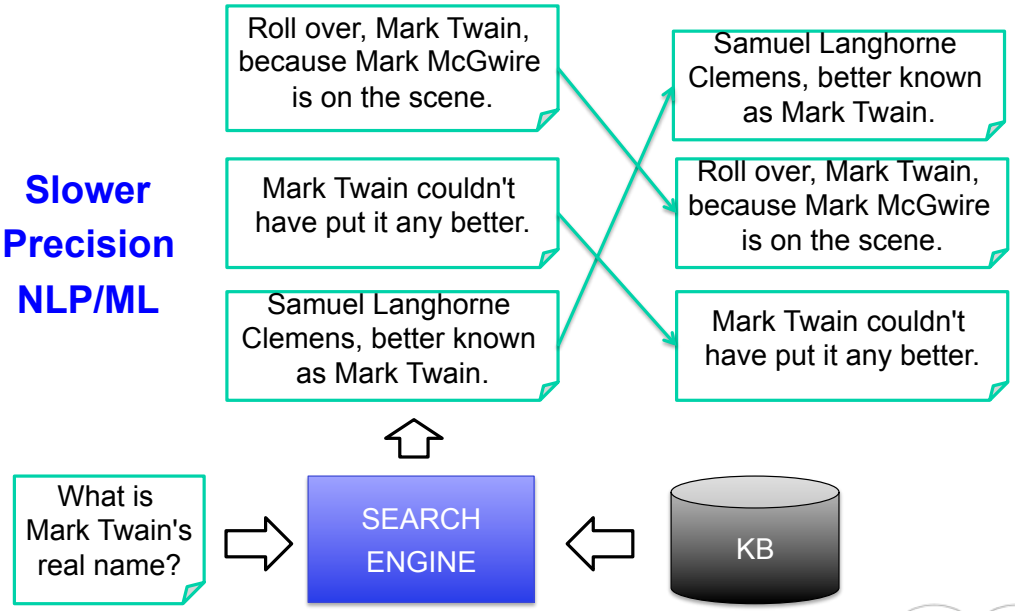
Samuel Langhorne Clemens, better known as Mark Twain.

What is Mark Twain's real name?

SEARCH ENGINE

KB

# Passage Reranking

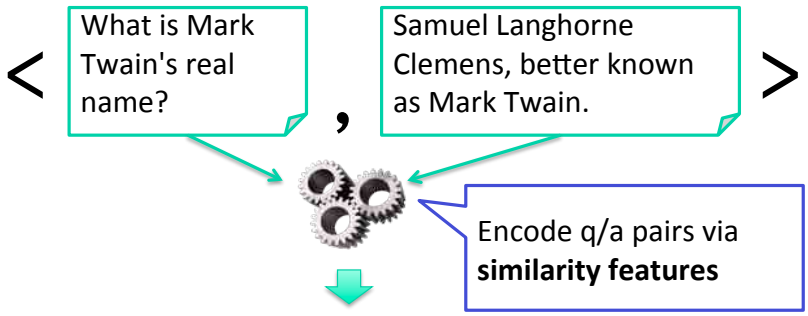



# Encoding question/answer pairs

< What is Mark Twain's real name? , Samuel Langhorne Clemens, better known as Mark Twain. >

< What is Mark Twain's real name? , Roll over, Mark Twain, because Mark McGwire is on the scene. >

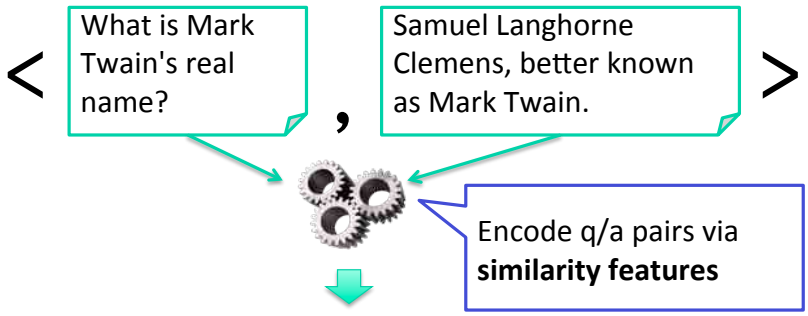
# Encoding question/answer pairs




(0.5, 0.4, 0.3, 0.0, 0.2,…, 1.0)

**lexical:** n-grams, Jaccard sim., etc.
**syntactic**: dependency path, TED
**semantic**: WN path, ESA, etc.

# Encoding question/answer pairs

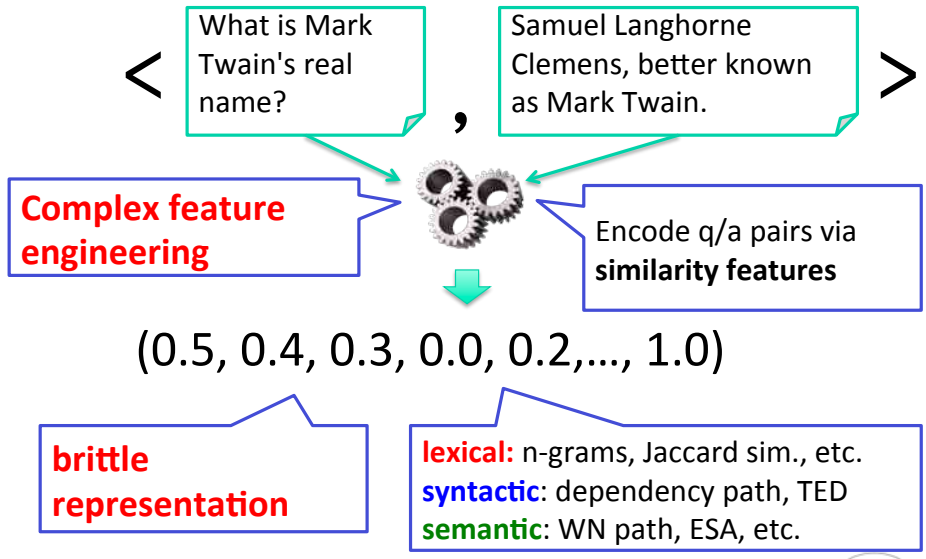



(0.5, 0.4, 0.3, 0.0, 0.2,…, 1.0)

**brittle representation**

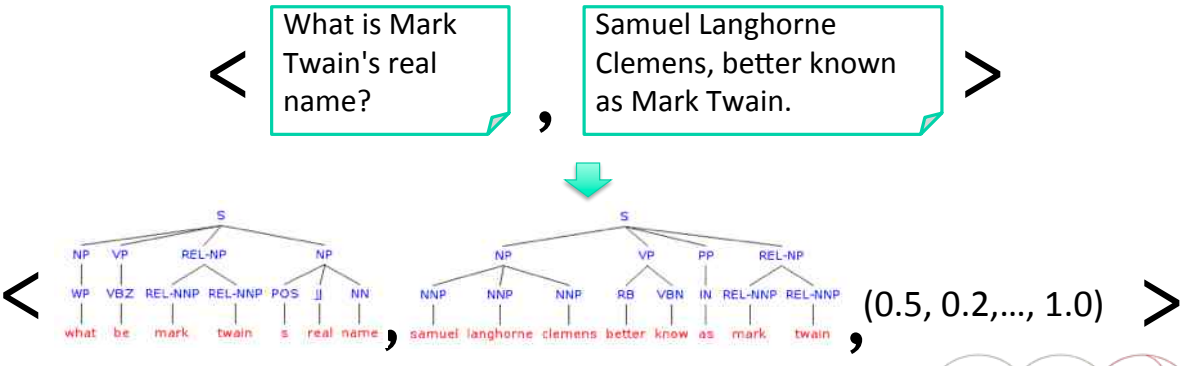
**lexical:** n-grams, Jaccard sim., etc.
**syntactic**: dependency path, TED
**semantic**: WN path, ESA, etc.

# Encoding question/answer pairs




# Our approach

- Model q/a pairs explicitly as linguistic structures
- Rely on Kernel Learning to automatically extract and learn powerful syntactic patterns



## Part I – Introduction to Structural kernels

- Classification function of kernel machines
- Kernel Definition (Kernel Trick)
- Kernel Operators
- String, Syntactic Tree Kernel, Partial Tree kernel (PTK)
- Efficiency

## Classification function of Kernel Machines

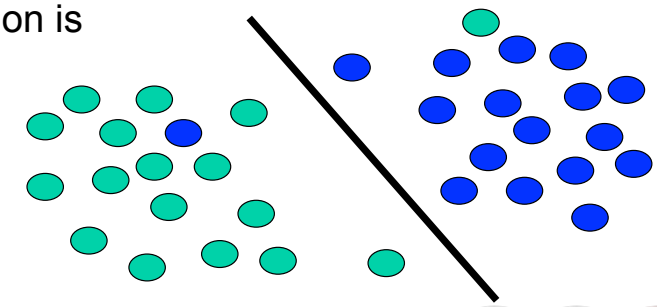
- The equation of a hyperplane is

$$f(\vec{x}) = \vec{x} \cdot \vec{w} + b = 0, \quad \vec{x}, \vec{w} \in \Re^n, b \in \Re$$

- $\vec{x}$ is the vector representing the classifying example
- $\vec{w}$ is the gradient of the hyperplane (learned model)
- The classification function is

$$h(\vec{x}) = \text{sign}(f(\vec{x}))$$

Note that the hyperplane classifier is just: $\vec{x} \cdot \vec{w} > -b$



## Kernel Trick

- Kernel Machines (e.g., SVMs or perceptron) are such that

$$\vec{w} = \sum_{j=1..\ell} \alpha_j y_j \vec{x}_j$$

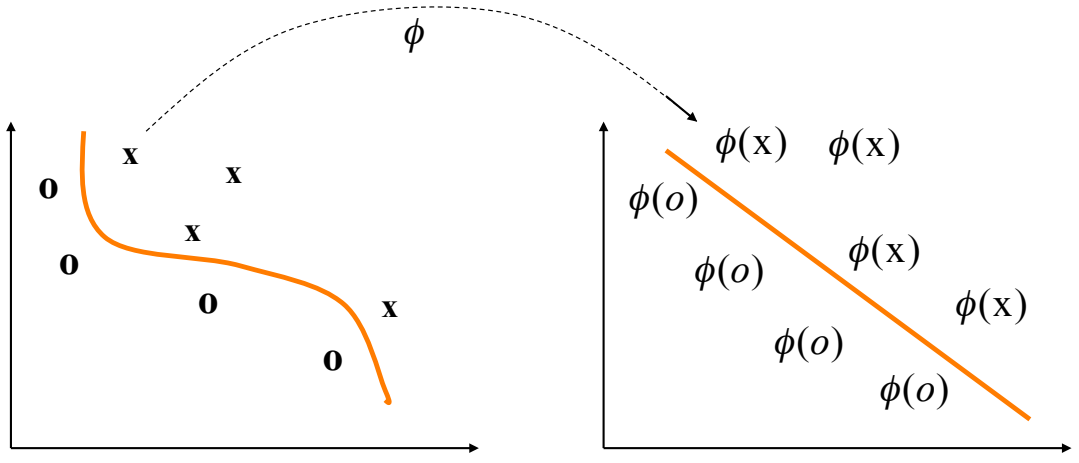
- Hence the classification function results:

$$\text{sgn}(\vec{w} \cdot \vec{x} + b) = \text{sgn}\left( \sum_{j=1..\ell} \alpha_j y_j \vec{x}_j \cdot \vec{x} + b \right)$$

- Note that data only appears in the scalar product

## The main idea of Kernel Functions

- Mapping vectors in a space where they are linearly separable, $\vec{x} \rightarrow \phi(\vec{x}) \quad \vec{w} \rightarrow \phi(\vec{w})$




## Classifying in the $\phi$ space

- In the space $\phi$, we can rewrite the classification function as:

$$h(\vec{x}) = \text{sgn}(\phi(\vec{w}) \cdot \phi(\vec{x}) + b_\phi) =$$

$$= \text{sgn}\left(\phi\left(\sum_{j=1..\ell} \alpha_j y_j \vec{x}_j\right) \cdot \phi(\vec{x}) + b_\phi\right) =$$

$$= \text{sgn}\left(\sum_{j=1..\ell} \alpha_j y_j \phi(\vec{x}_j) \cdot \phi(\vec{x}) + b_\phi\right) =$$

$$= \text{sgn}\left(\sum_{i=1..\ell} \alpha_j y_j k(\vec{x}_j, \vec{x}) + b_\phi\right)$$

## Kernel Function Definition

**Def. 2.26** *A kernel is a function $k$, such that* $\forall\ \vec{x}, \vec{z} \in X$

$$k(\vec{x}, \vec{z}) = \phi(\vec{x}) \cdot \phi(\vec{z})$$

*where $\phi$ is a mapping from $X$ to an (inner product) feature space.*

- Kernels are the product of mapping functions such as

$$\vec{x} \in \Re^n, \quad \vec{\phi}(\vec{x}) = (\phi_1(\vec{x}), \phi_2(\vec{x}), ..., \phi_m(\vec{x})) \in \Re^m$$

## Valid Kernel operations

- $k(x,z) = k_1(x,z)+k_2(x,z)$
- $k(x,z) = k_1(x,z)*k_2(x,z)$
- $k(x,z) = \alpha\, k_1(x,z)$
- $k(x,z) = f(x)f(z)$
- $k(x,z) = x'Bz$
- $k(x,z) = k_1(\phi(x),\phi(z))$

## Object Transformation [Moschitti et al, CLJ 2008]

- $$K(O_1,O_2) = \phi(O_1)\cdot\phi(O_2) = \phi_E(\phi_M(O_1))\cdot\phi_E(\phi_M(O_2)) = \phi_E(S_1)\cdot\phi_E(S_2) = K_E(S_1,S_2)$$
- ***Canonical Mapping***, $\phi_M()$
  - object transformation,
  - e. g., a syntactic parse tree into a verb subcategorization frame tree.
- ***Feature Extraction***, $\phi_E()$
  - maps the canonical structure in all its fragments
  - different fragment spaces, e.g. String and Tree Kernels

## Syntactic and Partial Tree Kernels

- Linear Kernels
- String and Word Sequence Kernels
- Syntactic Tree Kernel (STK)
- Partial Tree kernel (PTK)

## Linear Kernel

- In Text Categorization documents are word vectors

$$\Phi(d_x) = \vec{x} = (0,..,1,..,0,..,0,..,1,..,0,..,0,..,1,..,0,..,0,..,1,..,0,..,1)$$

buy market sell stocks trade

$$\Phi(d_z) = \vec{z} = (0,..,1,..,0,..,1,..,0,..,0,..,0,..,1,..,0,..,0,..,1,..,0,..,0)$$

buy company sell stock

- The dot product $\vec{x} \cdot \vec{z}$ counts the number of features in common
- This provides a sort of *similarity*

## String Kernel

- Given two strings, the number of matches between their substrings is evaluated
- E.g. Bank and Rank
  - B, a, n, k, Ba, Ban, Bank, Bk, an, ank, nk,..
  - R, a , n , k, Ra, Ran, Rank, Rk, an, ank, nk,..
- String kernel over sentences and texts
- Huge space but there are efficient algorithms

## Using character sequences

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

bank ank bnk bk b

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

rank ank rnk rk r

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

$$\vec{x} \cdot \vec{z} = \phi("bank") \cdot \phi("rank") = k("bank","rank")$$

## Efficient Evaluation: Intuition

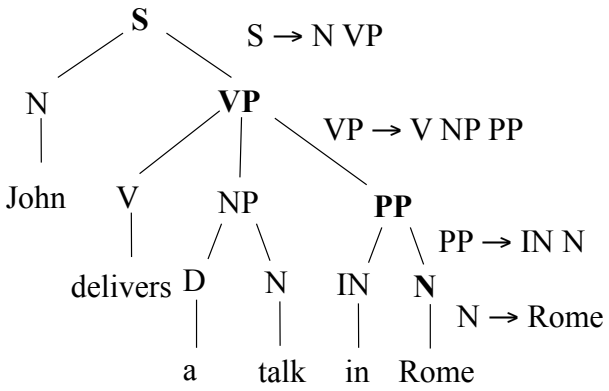
- Dynamic Programming technique over:
  - The size of the two input strings, *m*, *n* and
  - The size of their common substrings, *p*
- Evaluate the spectrum string kernels
  - Substrings of size *p*
- Sum the contribution of the different *p* spectra

## Tree kernels

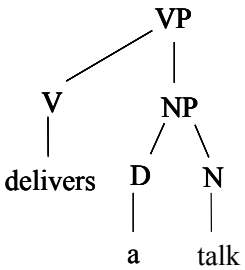
- Syntactic Tree Kernel (STK)
- Partial Tree kernel (PTK)
- Efficient computation

# Example of a parse tree

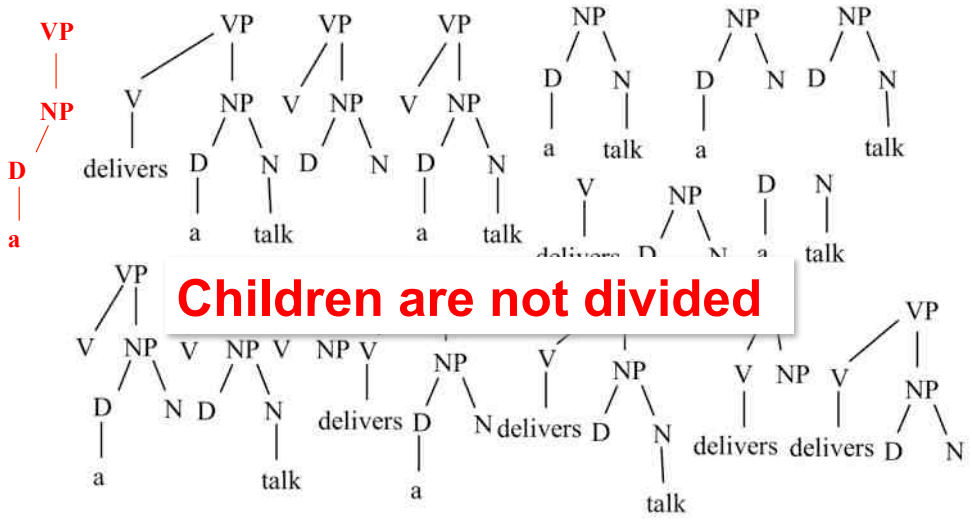
- "John delivers a talk in Rome"




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

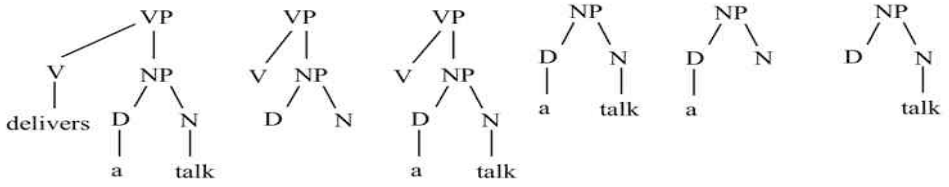


## The overall fragment set

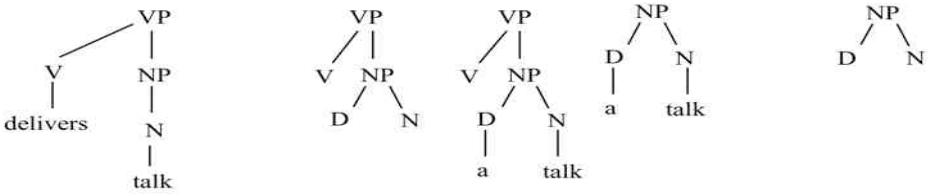



## Explicit kernel space

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



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



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

## Efficient evaluation of the scalar product: Syntactic Tree Kernel (STK)

$$\vec{x} \cdot \vec{z} = \phi(T_x) \cdot \phi(T_z) = K(T_x, T_z) =$$
$$= \sum_{n_x \in T_x} \sum_{n_z \in T_z} \Delta(n_x, n_z)$$

## Efficient evaluation of the scalar product: Syntactic Tree Kernel (STK)

$$\vec{x} \cdot \vec{z} = \phi(T_x) \cdot \phi(T_z) = K(T_x, T_z) =$$
$$= \sum_{n_x \in T_x} \sum_{n_z \in T_z} \Delta(n_x, n_z)$$

- [Collins and Duffy, ACL 2002] evaluate $\Delta$ in $O(n^2)$:

$$\Delta(n_x, n_z) = 0, \quad \text{if the productions are different else}$$
$$\Delta(n_x, n_z) = 1, \quad \text{if pre-terminals else}$$
$$\Delta(n_x, n_z) = \prod_{j=1}^{nc(n_x)} (1 + \Delta(ch(n_x, j), ch(n_z, j)))$$

## 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

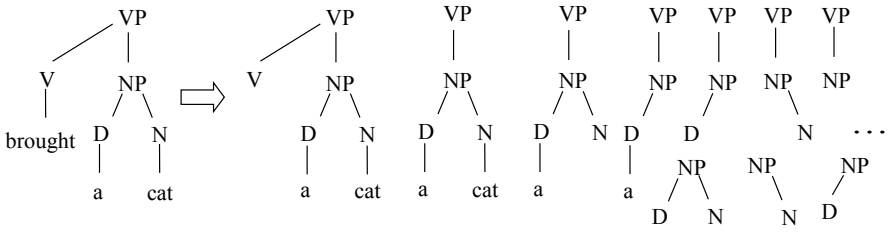
$$K'(T_x, T_z) = \frac{K(T_x, T_z)}{\sqrt{K(T_x, T_x) \times K(T_z, T_z)}}$$

## Observations

- We can order the production rules used in $T_x$ and $T_z$, at loading time
- At learning time we can evaluate NP in $|T_x| + |T_z|$ *running time* [Moschitti, EACL 2006]
- If $T_x$ and $T_z$ are generated by only one production rule ⇒ $O(|T_x| \times |T_z|)$ … ***Very Unlikely!!!!***

# Partial Tree Kernel (PTK)
**[Moschitti, ECML 2006]**

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




# Partial Tree Kernel - Definition

- 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:

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

## Efficient Evaluation (1)

- In [Taylor and Cristianini, 2004 book], sequence kernels with weighted gaps are factorized with respect to different subsequence sizes
- We treat children as sequences and apply the same theory

$$\Delta(n_1, n_2) = \mu\left(\lambda^2 + \sum_{p=1}^{lm} \Delta_p(c_{n_1}, c_{n_2})\right)$$

Given the two child sequences $s_1 a = c_{n_1}$ and $s_2 b = c_{n_2}$ ($a$ and $b$ are the last children), $\Delta_p(s_1 a, s_2 b) =$

$$\Delta(a,b) \times \boxed{\sum_{i=1}^{|s_1|} \sum_{r=1}^{|s_2|} \lambda^{|s_1|-i+|s_2|-r} \times \Delta_{p-1}(s_1[1:i], s_2[1:r])}$$

$D_p$

## Efficient Evaluation (2)

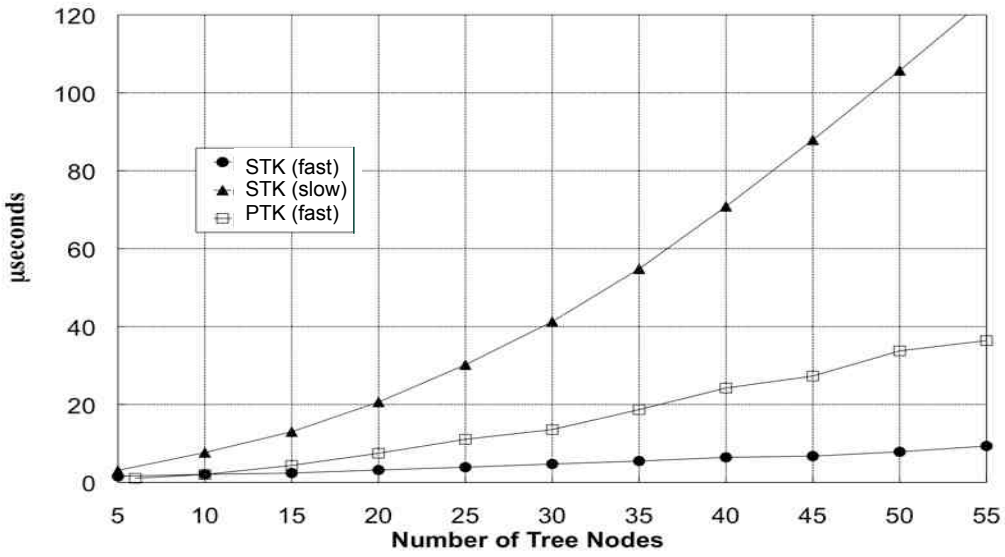
$$\Delta_p(s_1 a, s_2 b) = \begin{cases} \Delta(a,b) D_p(|s_1|, |s_2|) \text{ if } a = b; \\ 0 \qquad\qquad\qquad\qquad otherwise. \end{cases}$$

Note that $D_p$ satisfies the recursive relation:

$$\begin{aligned} D_p(k,l) \quad &= \Delta_{p-1}(s_1[1:k], s_2[1:l]) + \lambda D_p(k, l-1) \\ &+ \lambda D_p(k-1, l) + \lambda^2 D_p(k-1, l-1). \end{aligned}$$

- The complexity of finding the subsequences is $O(p|s_1||s_2|)$
- Therefore the overall complexity is $O(p\rho^2|N_{T_1}||N_{T_2}|)$ where $\rho$ is the maximum branching factor ($p = \rho$)

# Running Time of Tree Kernel Functions

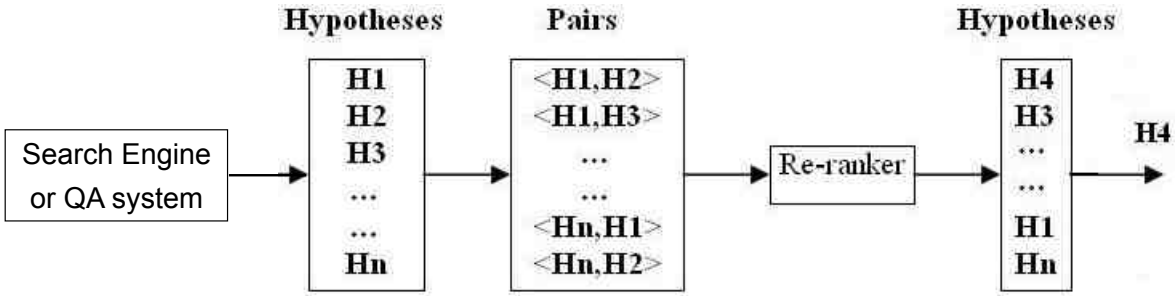



# Outline: Kernels for Ranking

- Reranking with kernels
  - Preference Kernel
  - Reranking Passages with relational representations
  - Shallow Syntax + semantic information
  - Dependency Trees
  - Semantic Roles
  - Discourse
  - Link Open Data

# Relational Kernels for Passage Reranking

## Preference Reranking for documents/ passages




- The initial rank is provided by a search engine (or also a powerful QA system)
- New idea: a boost can be achieved by capturing the relation between question and answer passage

## More formally

- Build a set of hypotheses: Q and A pairs
- These are used to build pairs of 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

## Preference Reranking Kernel

$H_1 > H_2$ and $H_3 > H_4$ then consider training vectors:

$\vec{Z}_1 = \phi(H_1) - \phi(H_2)$ and $\vec{Z}_2 = \phi(H_3) - \phi(H_4) \Rightarrow$ the dot product is:

$$\vec{Z}_1 \bullet \vec{Z}_2 = \left(\phi(H_1) - \phi(H_2)\right) \bullet \left(\phi(H_3) - \phi(H_4)\right) =$$

$$\phi(H_1) \bullet \phi(H_3) - \phi(H_1) \bullet \phi(H_4) - \phi(H_2) \bullet \phi(H_3) + \phi(H_2) \bullet \phi(H_4)$$

$$= K(H_1, H_3) - K(H_1, H_4) - K(H_2, H_3) + K(H_2, H_4)$$

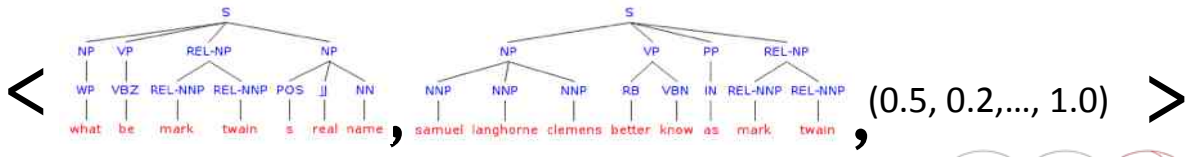
Let $H_i = \langle q_i, a_i \rangle$, $H_j = \langle q_j, a_j \rangle$

$$K(H_i, H_j) = PTK(q_i, q_j) + PTK(a_i, a_j)$$

# Our approach

- Model q/a pairs explicitly as linguistic structures
- Rely on Kernel Learning to automatically extract and learn powerful syntactic patterns

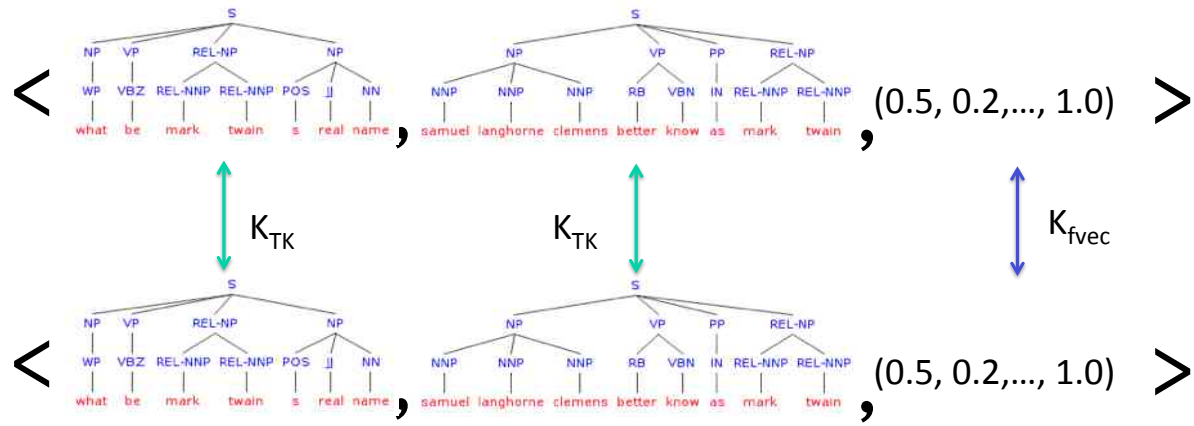
< What is Mark Twain's real name? , Samuel Langhorne Clemens, better known as Mark Twain. >




(0.5, 0.2,..., 1.0) >



# Computing kernel between q/a pairs




< (0.5, 0.2,..., 1.0) >

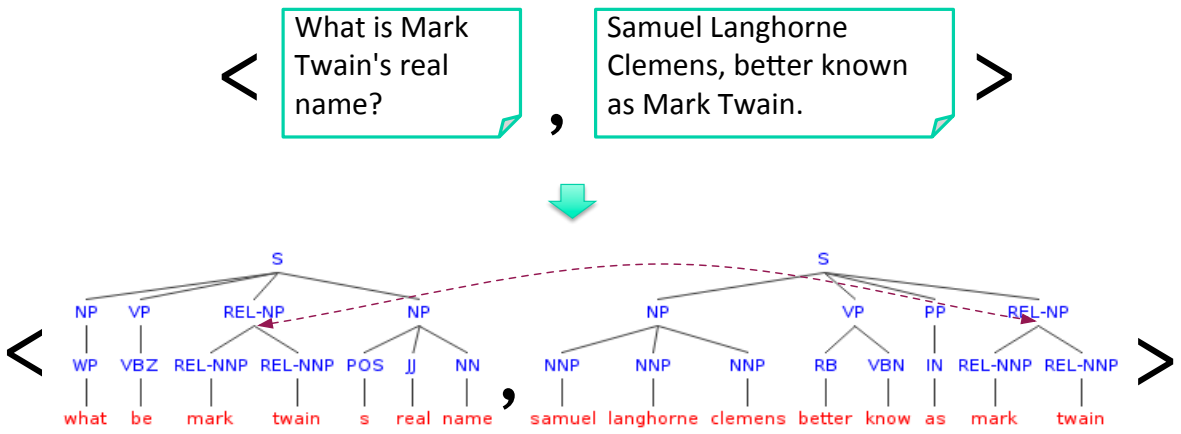
$K_{TK}$ $K_{TK}$ $K_{fvec}$

< (0.5, 0.2,..., 1.0) >

# Structural representations of q/a pairs

- NLP structures are rich sources of features
  - Shallow syntactic and dependency trees
- Linking related fragments between question and answer is important:
  - Simple lemma matching (Severyn and Moschitti, 2012)
  - Semantic linking (Severyn et al., CoNLL, CIKM 2013)

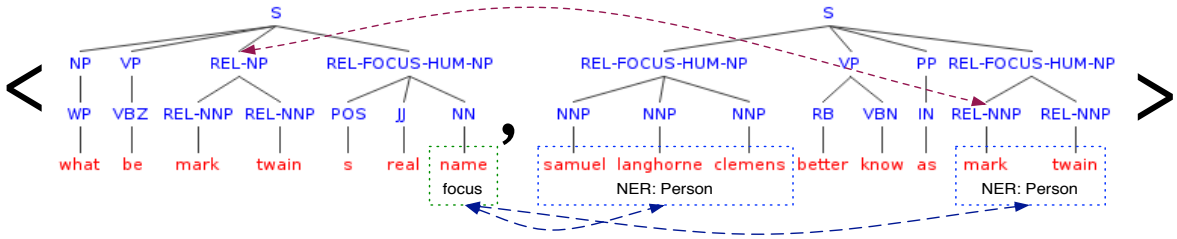
# Relational shallow trees
## (Severyn and Moschitti, 2012)




# Semantic linking
## (Severyn et al., 2013)

Find question category (QC):

**HUM**



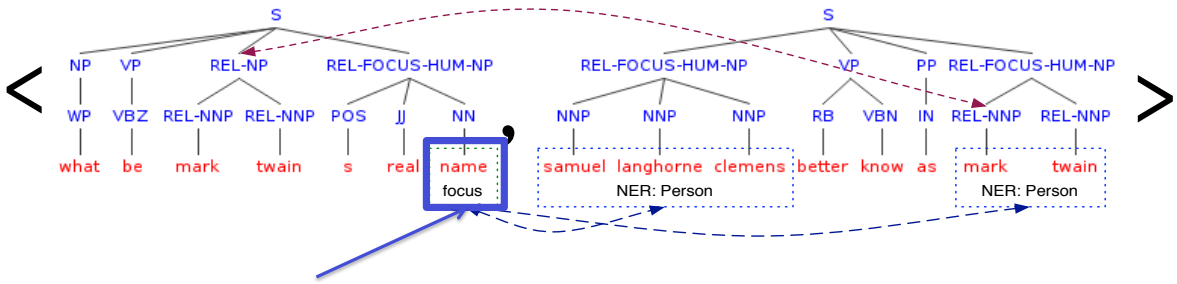





# Semantic linking

Find question category (QC):

**HUM**

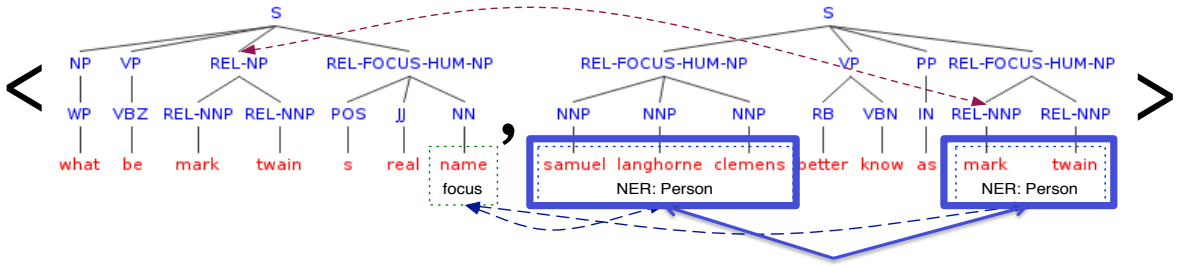



Find focus (FC):

**name**

# Semantic linking

Find question category (QC):
**HUM**



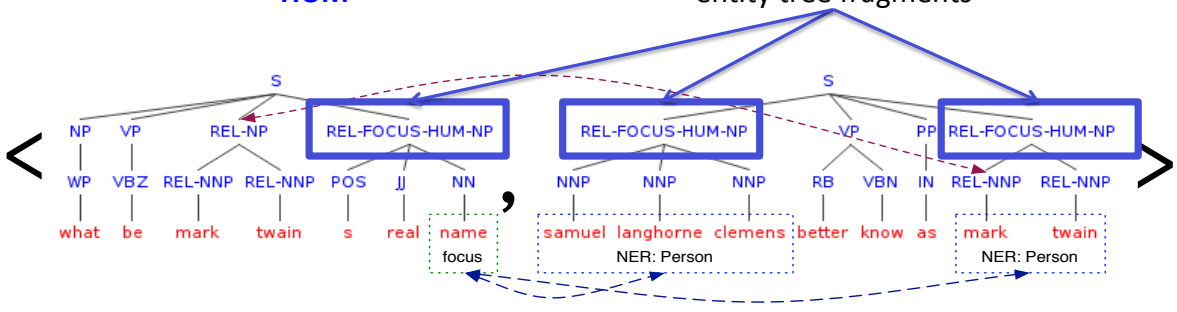

Find focus (FC):
**name**

Find entities according to question category in the answer passage (NER)

# Semantic linking

Find question category (QC):
**HUM**

Link focus word and named entity tree fragments




Find focus (FC):
**name**

Find entities according to question category in the answer passage (NER)

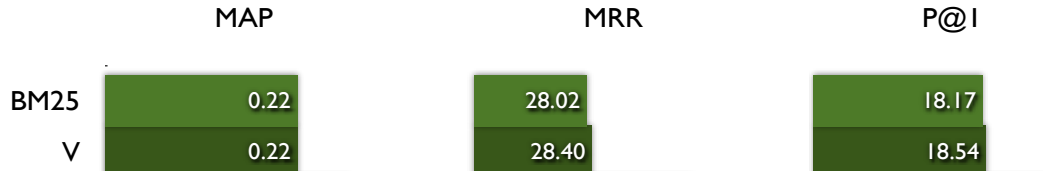
# Experiments and models

- Data
  - TREC QA 2002 & 2003 (824 questions)
- Systems
  - BM25 from IR
  - CH - shallow tree [Severyn & Moschitti, 2012]
  - V - similarity feature vector model
  - +FC+QC - semantic linking
  - +TFC+QC - typed semantic linking

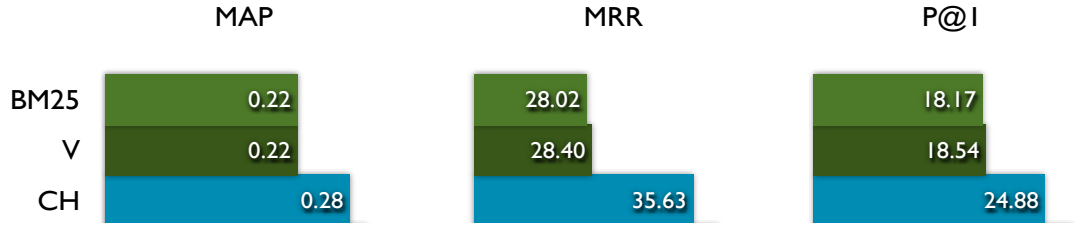
# Feature Vector Representation

- Lexical
  - Term-overlap: n-grams of lemmas, POS tags, dependency triplets
- Syntactic
  - Tree kernel score over shallow syntactic and dependency trees
- QA compatibility
  - Question category
  - NER relatedness – proportion of NER types related to the question category

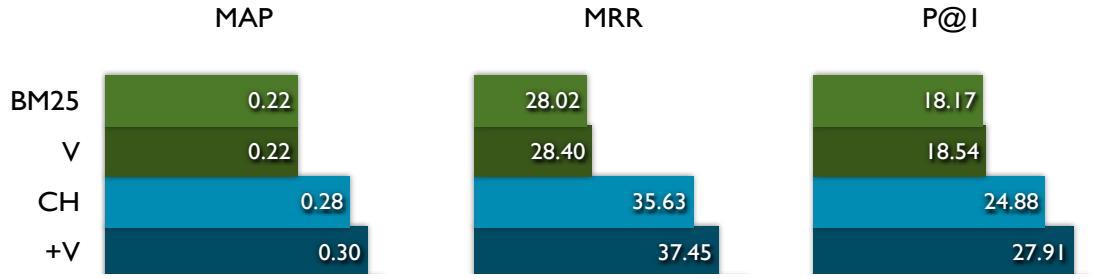
# Structural representations on TREC QA



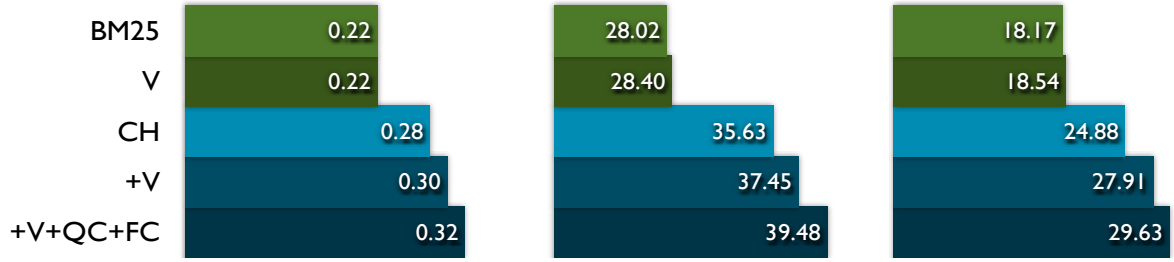

# Structural representations on TREC QA



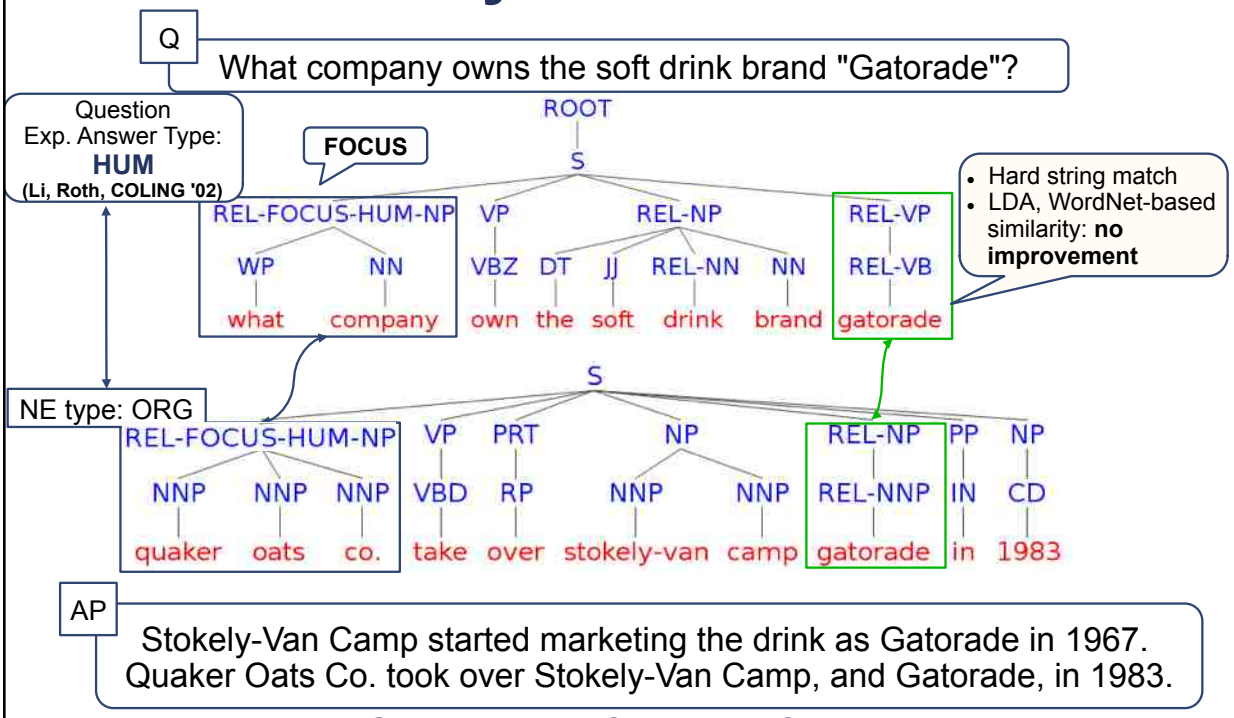
# Structural representations on TREC QA




# Structural representations on TREC QA

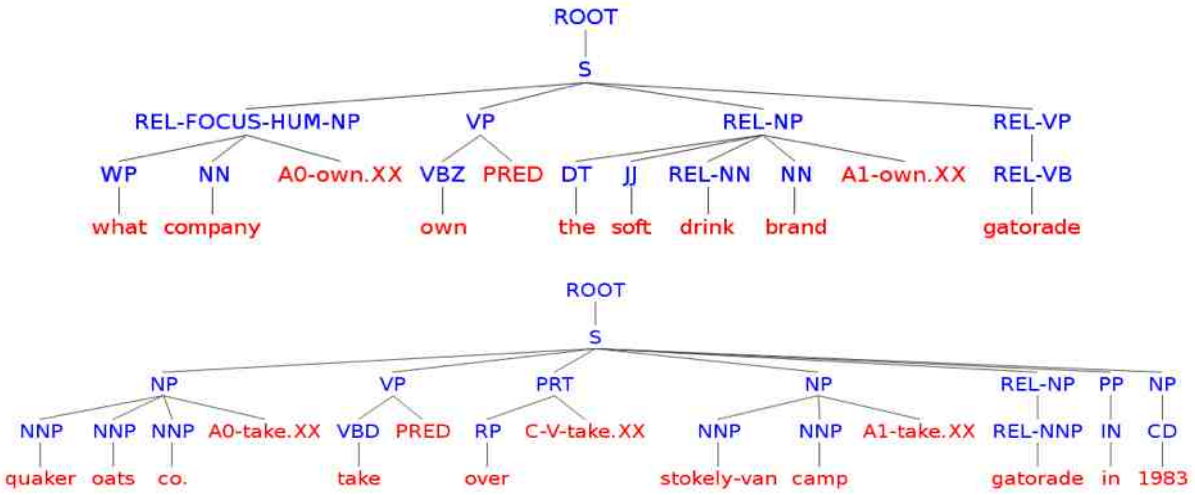


# Summary: CH+V+QC+TFC

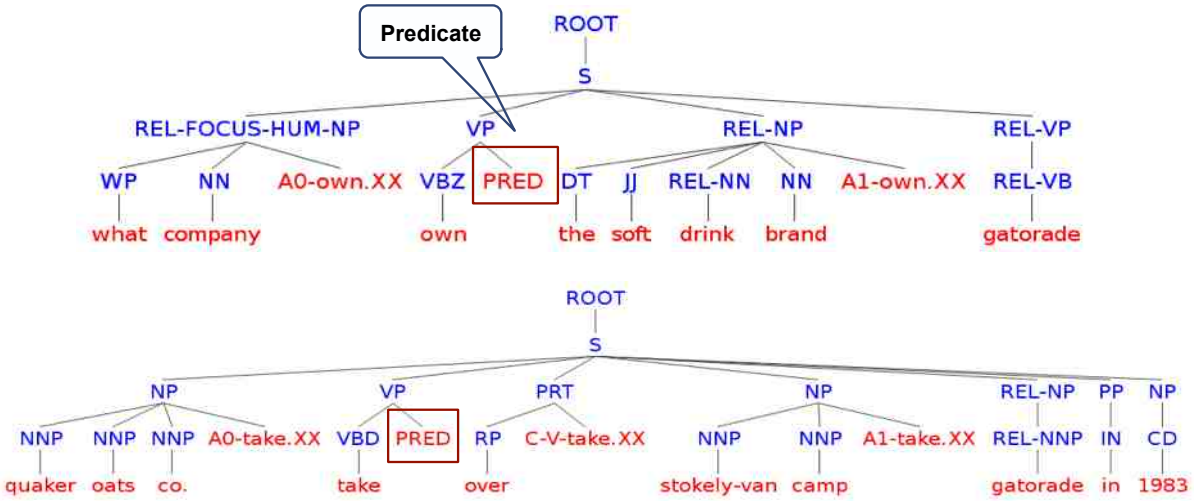



**(Severyn et. al, CoNLL '13, CIKM '13)**

# CH+V+QC+TFC+SRL

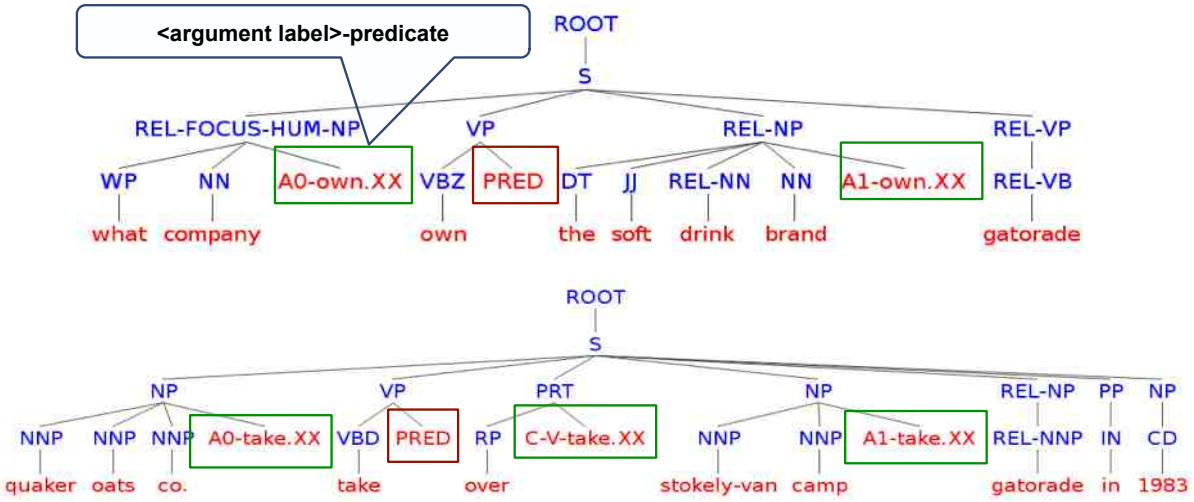


# CH+V+QC+TFC+SRL

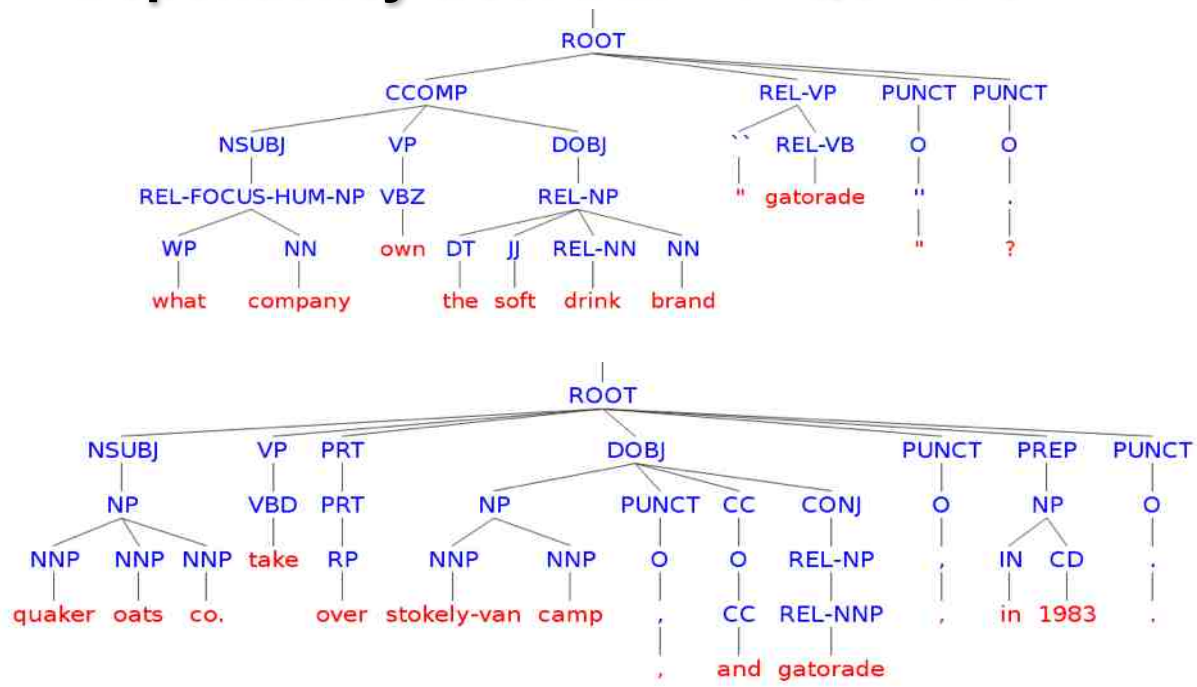




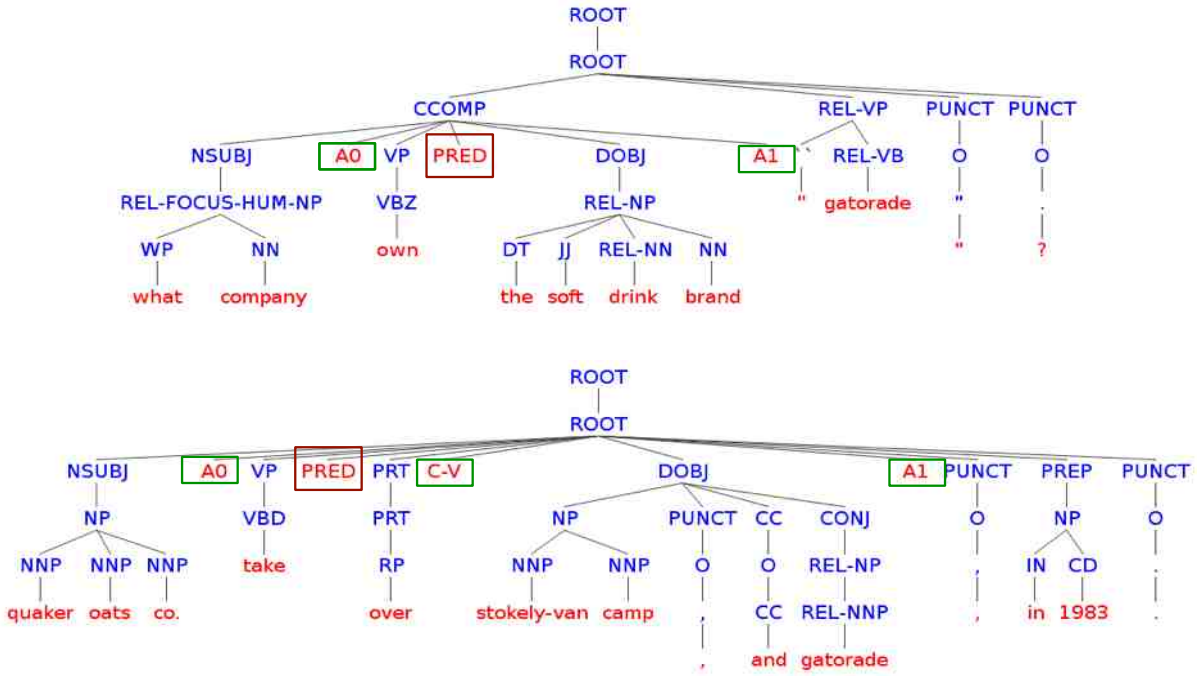

# CH+V+QC+TFC+SRL



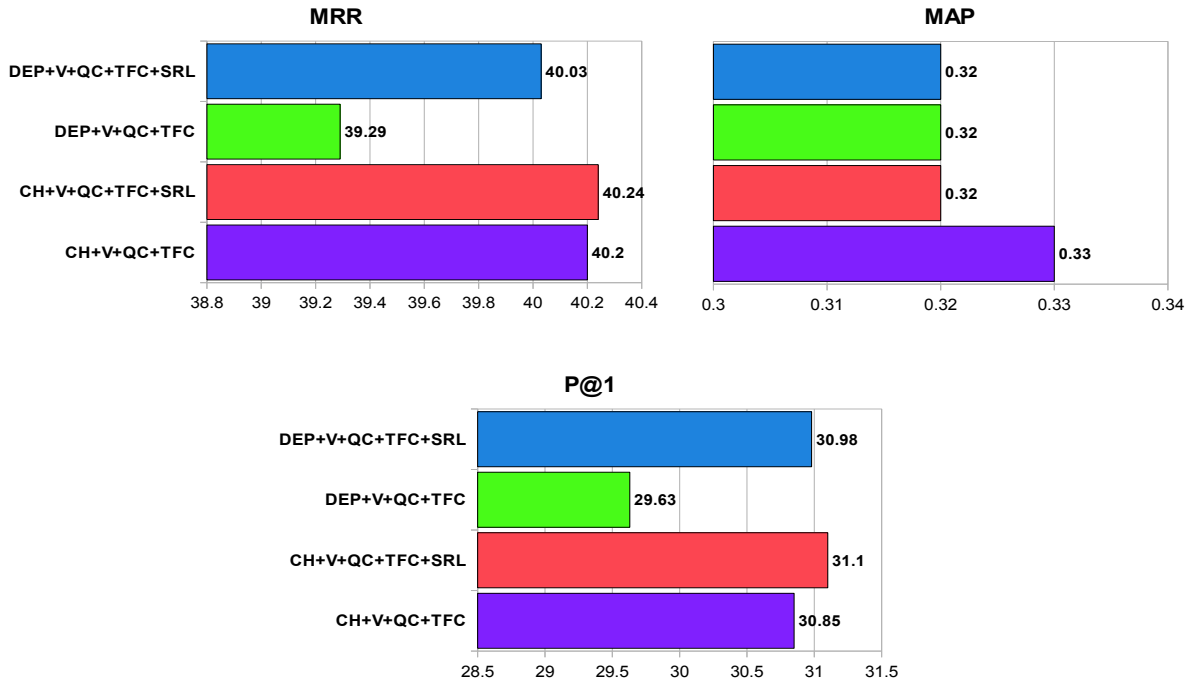

# Dependency tree: DEP+V+QC+TFC



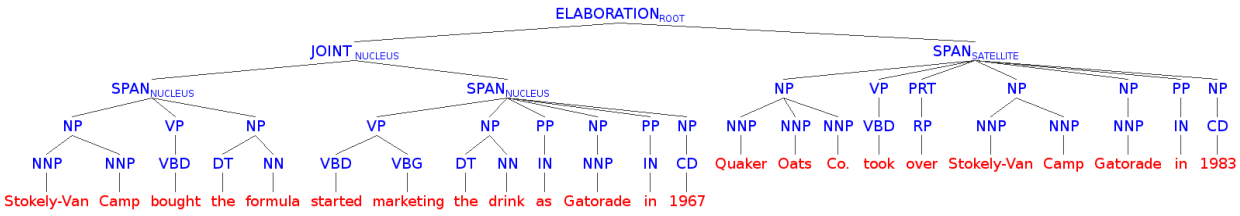

# DEP+V+QC+TFC+SRL



# Results




# Discourse?

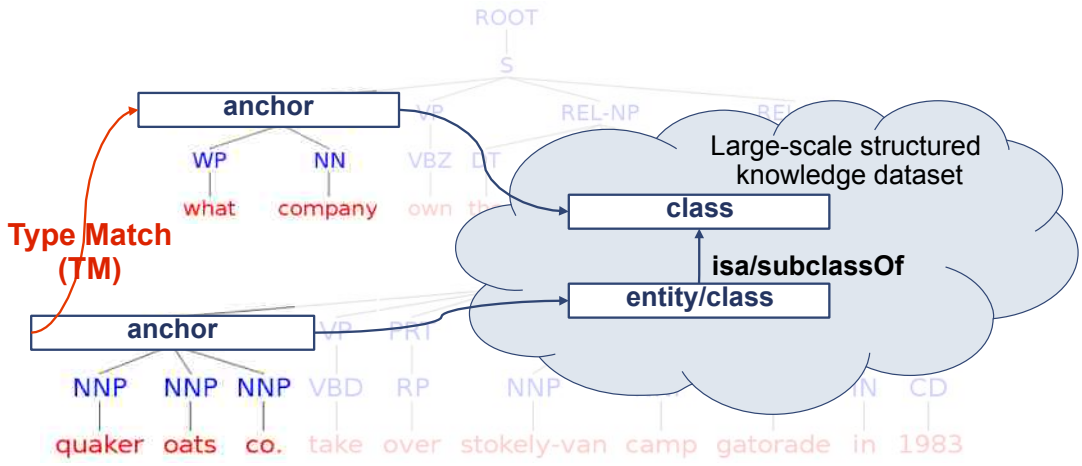


# Semantic Structures from Link Open Data

## Using Type Match Relation

Q

What company owns the soft drink brand "Gatorade"?




AP

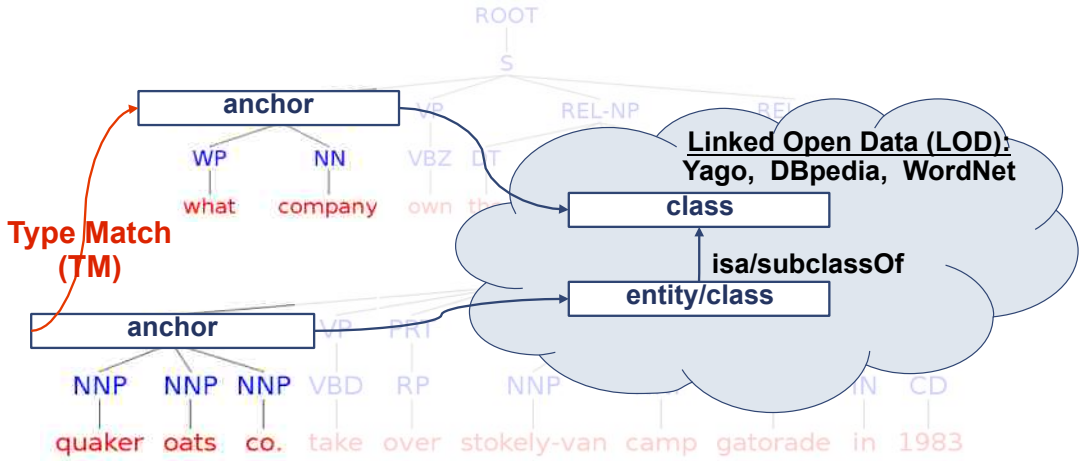
Stokely-Van Camp started marketing the drink as Gatorade in 1967.
Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.

**(Timoshenko et. al, EACL'14)**

# Using Type Match Relation

Q

What company owns the soft drink brand "Gatorade"?




AP

Stokely-Van Camp started marketing the drink as Gatorade in 1967. Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.

# Linked Open Data

- Structured knowledge published according to **LOD principles**
  - organized as directed graph/statements

# Linked Open Data

- Structured knowledge published according to **LOD principles**
  - organized as directed graph/statements
  - commonly shared knowledge schemes
    - **rdfs:subClassOf**, **rdf:type**, **rdf:label**

# Linked Open Data

- Structured knowledge published according to **LOD principles**
  - organized as directed graph/statements
  - commonly shared knowledge schemes
    - **rdfs:subClassOf**, **rdf:type**, **rdf:label**
- > 250 data sets
  - **DBpedia** (> 4 mln entities): extracted from **Wikipedia**
  - **YAGO** (> 10 mln entities): **Wikipedia + WordNet**

# Match algorithm

- **Input:** text passages Q, AP; LOD dataset

**Q** What company owns the soft drink brand "Gatorade"?

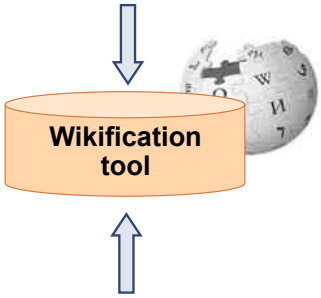
**AP** Stokely-Van Camp started marketing the drink as Gatorade in 1967. Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.

# Match algorithm

1. Detect anchors in AP
2. For each anchor extract references

**Q** What company owns the soft drink brand "Gatorade"?

If **LOD dataset = (YAGO OR DBpedia)**
- **YAGO, DBpedia** are aligned with **Wikipedia** on entity level




**AP** Stokely-Van Camp started marketing the drink as Gatorade in 1967. Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.

# Match algorithm

1. Detect anchors in AP
2. For each anchor extract references

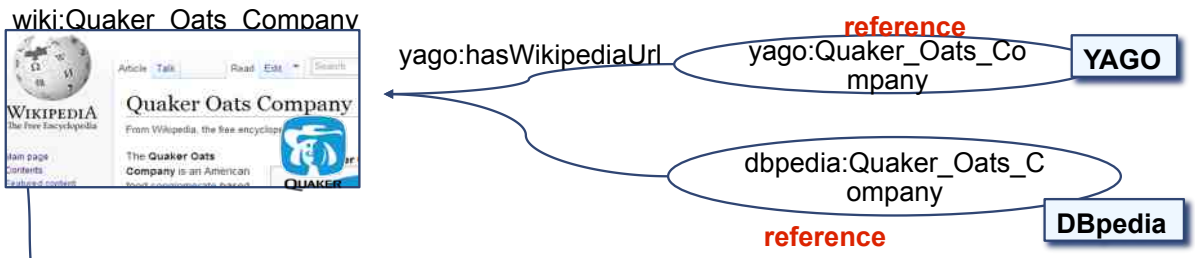
Q: What company owns the soft drink brand "Gatorade"?


AP: Stokely-Van Camp started marketing the drink as Gatorade in 1967.
Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.


# Match algorithm

1. Detect anchors in AP
2. For each anchor extract references

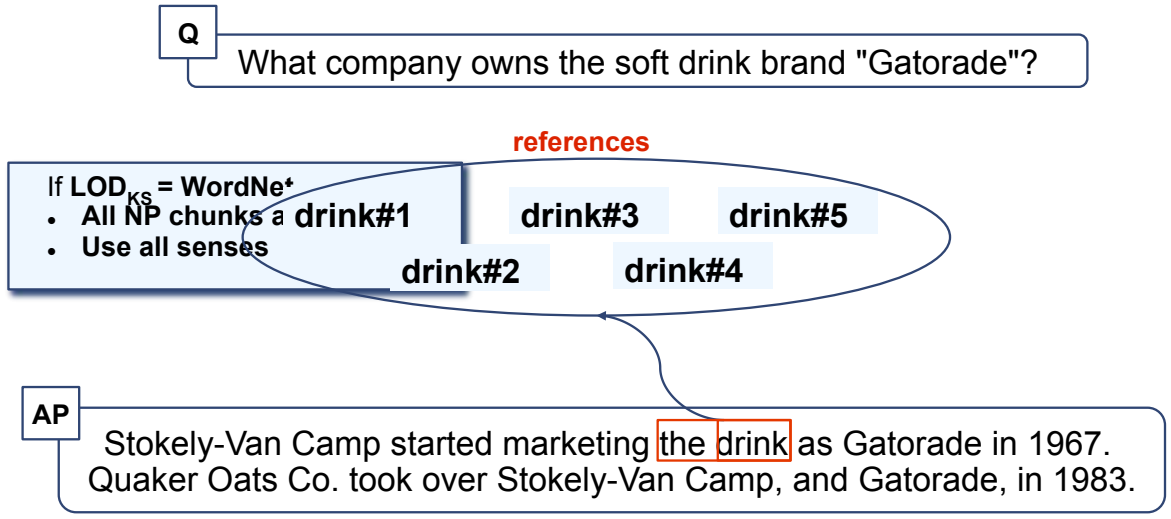
Q: What company owns the soft drink brand "Gatorade"?


AP: Stokely-Van Camp started marketing the drink as Gatorade in 1967.
Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.

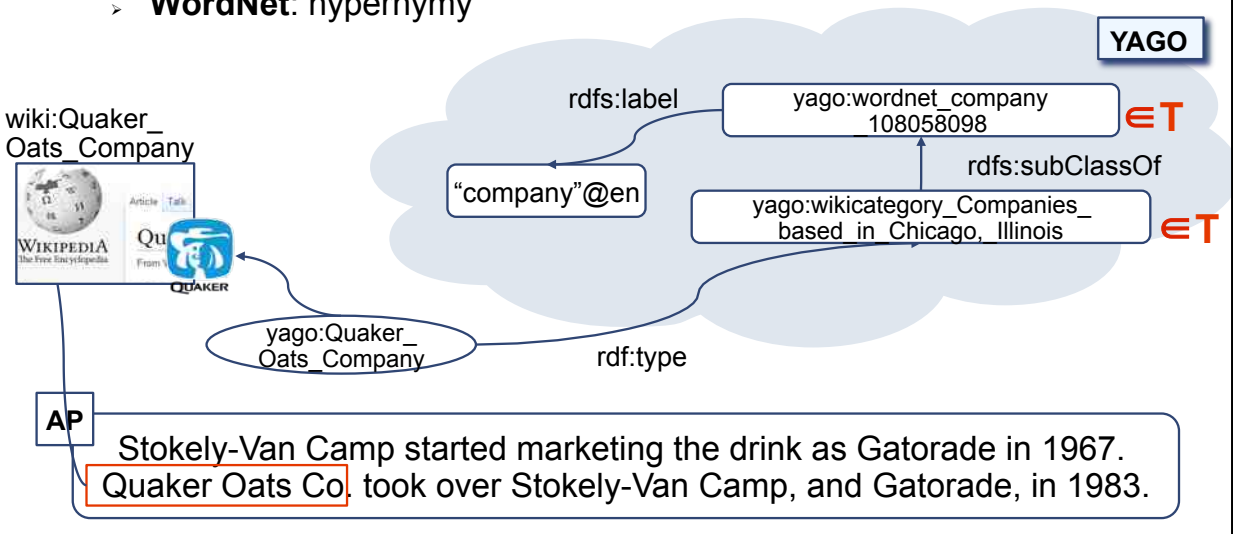
# Match algorithm

1. Detect anchors in AP
2. For each anchor extract references



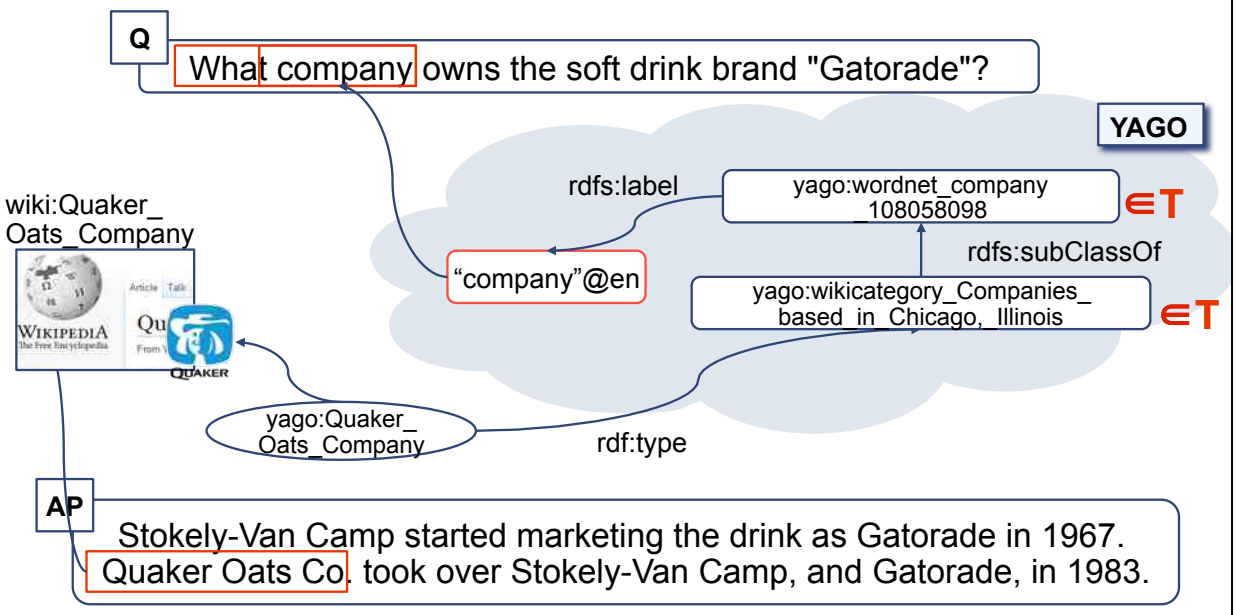

# Match algorithm

3. For each reference extract a set of types
   - **YAGO, DBpedia**: rdf:type, rdfs:subclassOf
   - **WordNet**: hypernymy



# Match algorithm

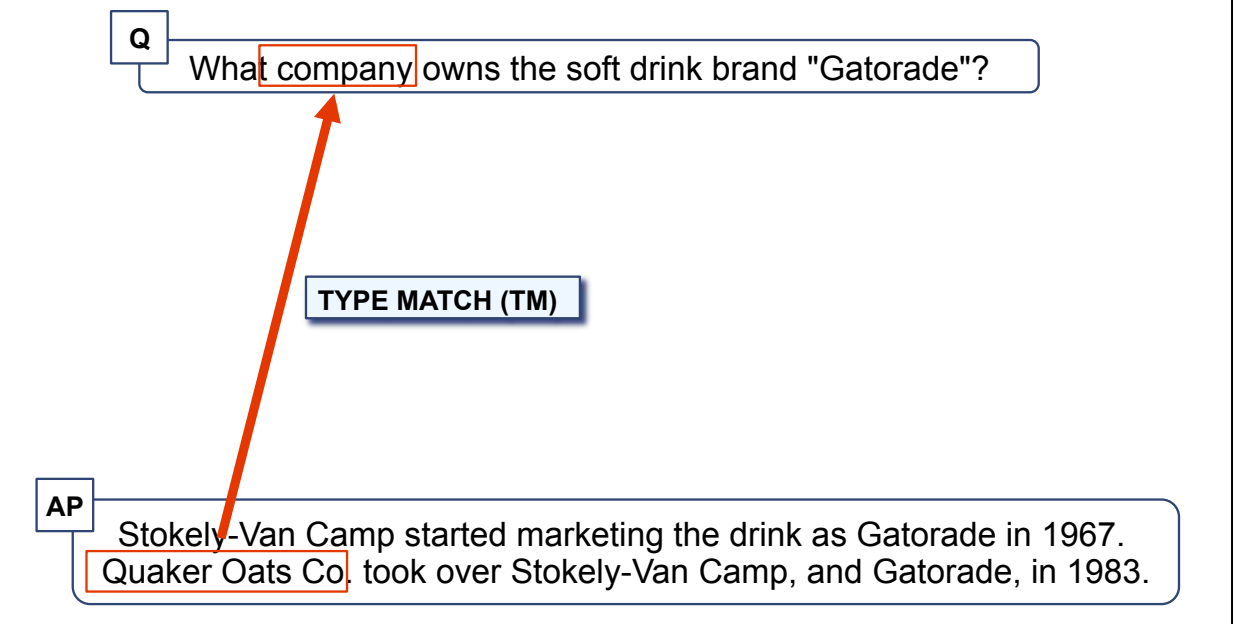
4. Match type names to NP chunks in **Q**





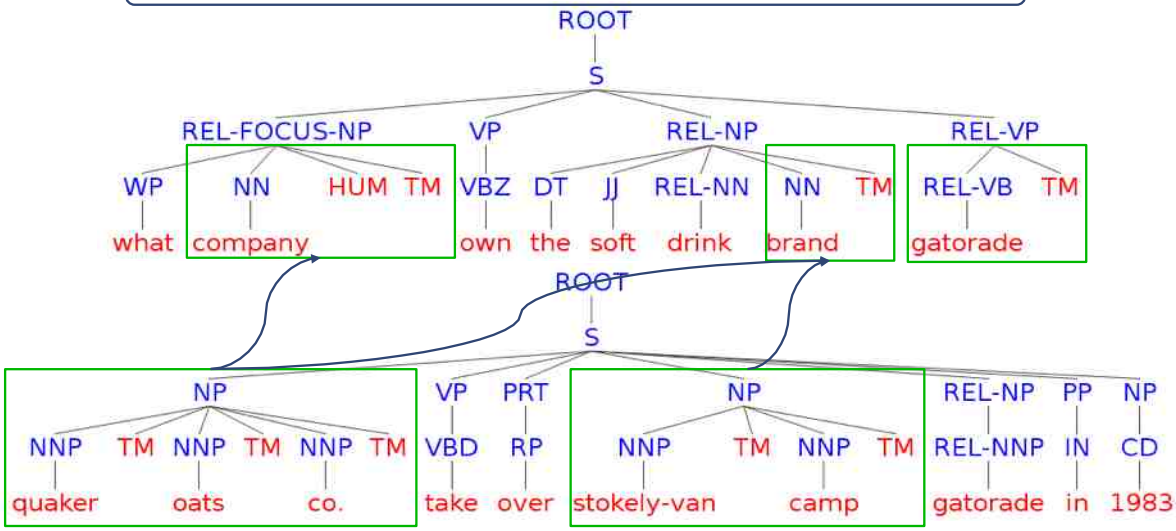
# Match algorithm

4. Match type names to NP chunks in **Q**

# Encoding type match: $TM_N$

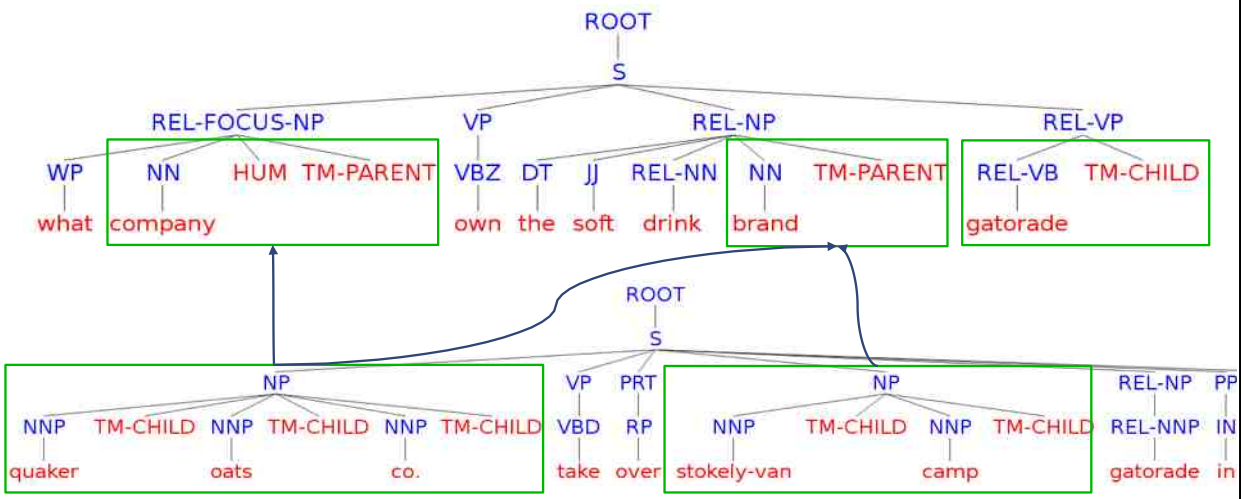
Q: What company owns the soft drink brand "Gatorade"?




AP: Stokely-Van Camp started marketing the drink as Gatorade in 1967. Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.

# Encoding type match: $TM_{ND}$

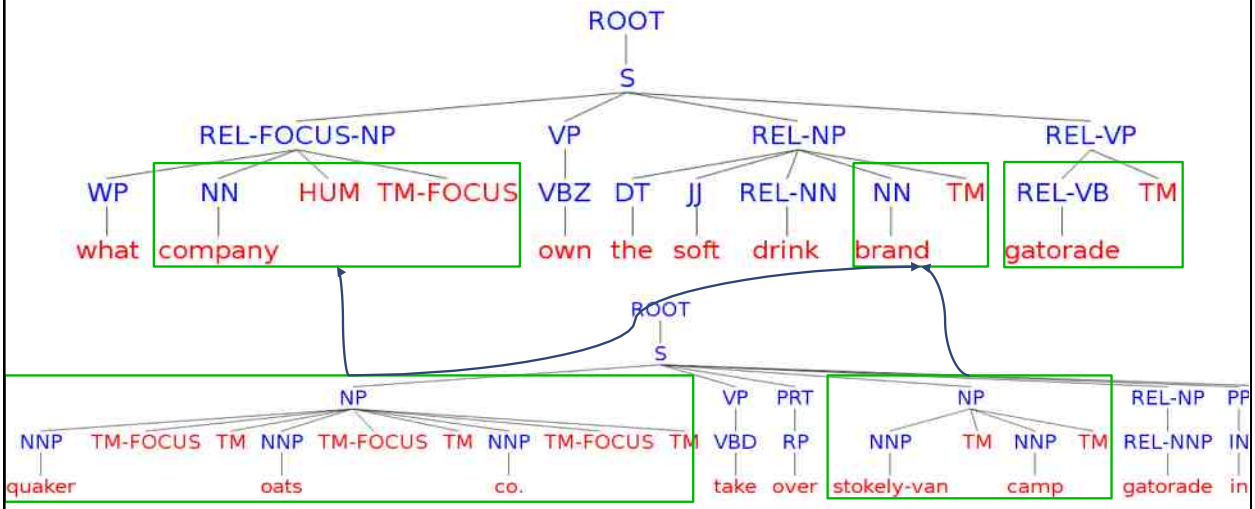
Q: What company owns the soft drink brand "Gatorade"?




AP: Stokely-Van Camp started marketing the drink as Gatorade in 1967. Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.

# Encoding type match: $TM_{NF}$

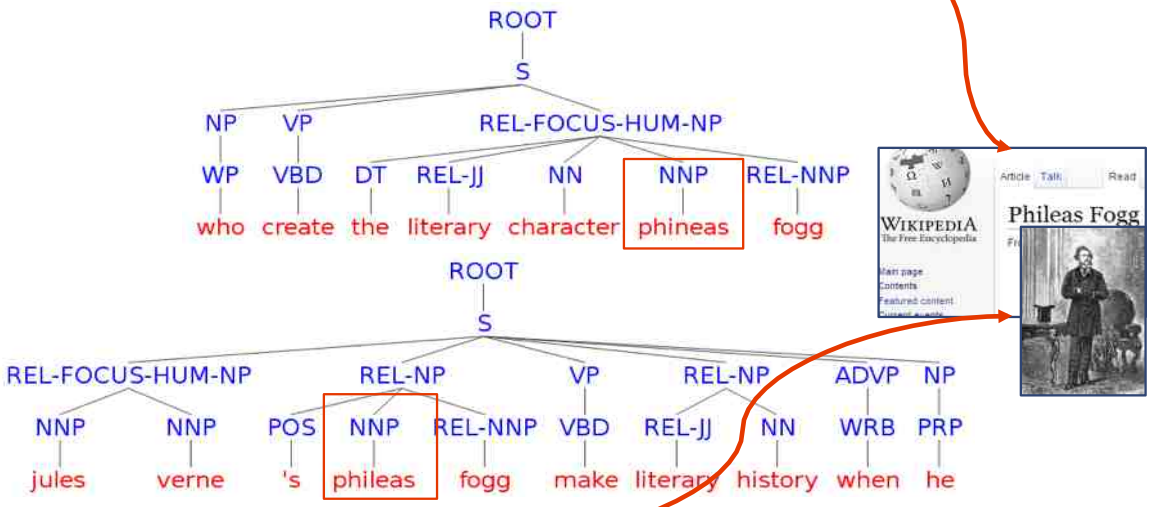
Q: What company owns the soft drink brand "Gatorade"?




AP: Stokely-Van Camp started marketing the drink as Gatorade in 1967. Quaker Oats Co. took over Stokely-Van Camp, and Gatorade, in 1983.

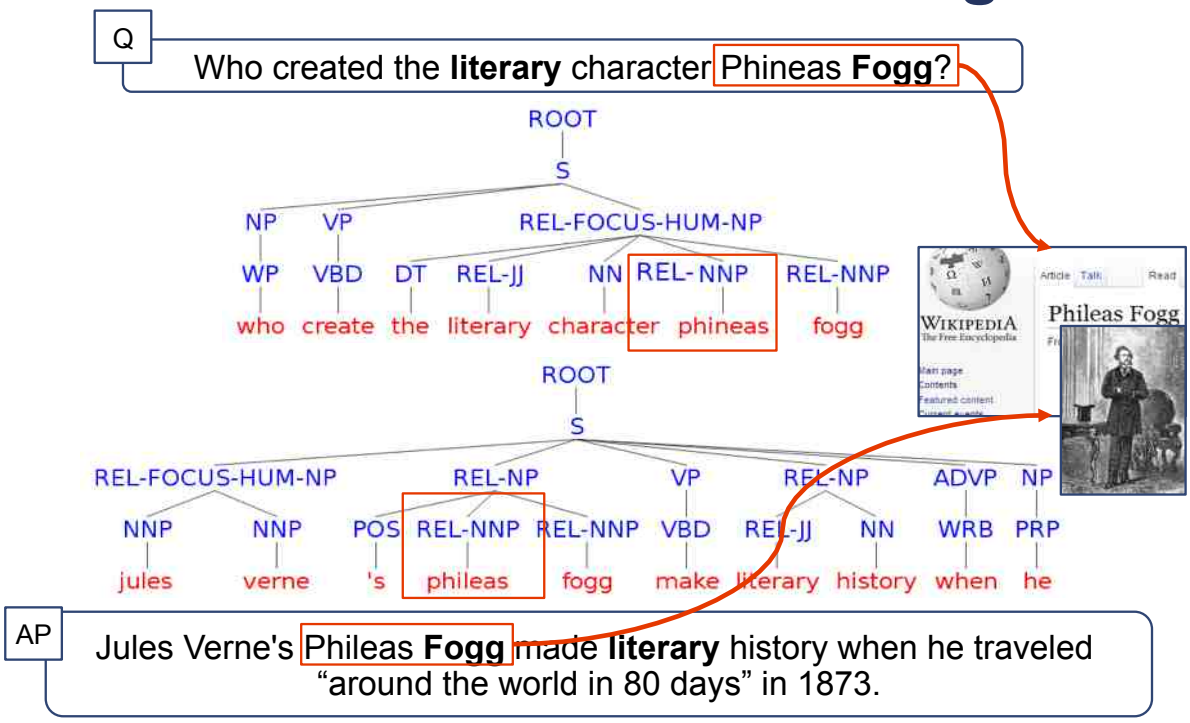
# Wiki-based REL-matching

Q: Who created the **literary** character Phineas **Fogg**?




AP: Jules Verne's Phileas **Fogg** made **literary** history when he traveled "around the world in 80 days" in 1873.

# Wiki-based REL-matching




# Experimental setting (1)

- **TREC QA 2002/2003** dataset
  - 824 factoid questions + answer patterns
- **AQUAINT** corpus for answer passage retrieval

# Experimental setting (1)

- **TREC QA 2002/2003** dataset
  - 824 factoid questions + answer patterns
- **AQUAINT** corpus for answer passage retrieval
- **5-fold** cross-validation
  - 165 questions for test, 649 questions for training
  - 10 answer passages per training question → 4800 examples/fold
  - 50 answer passages per test question → 8200 examples/fold

# Experimental setting (2)

- **Preference reranking** with kernels (Severyn et al, SIGIR '12)
  - **Partial Tree Kernel** for structures (Moschitti, ECML '06)
  - **polynomial** kernel for vectors
- Prune unrelated substructures
- Wikification
  - wikipediaminer
  - MachineLinking

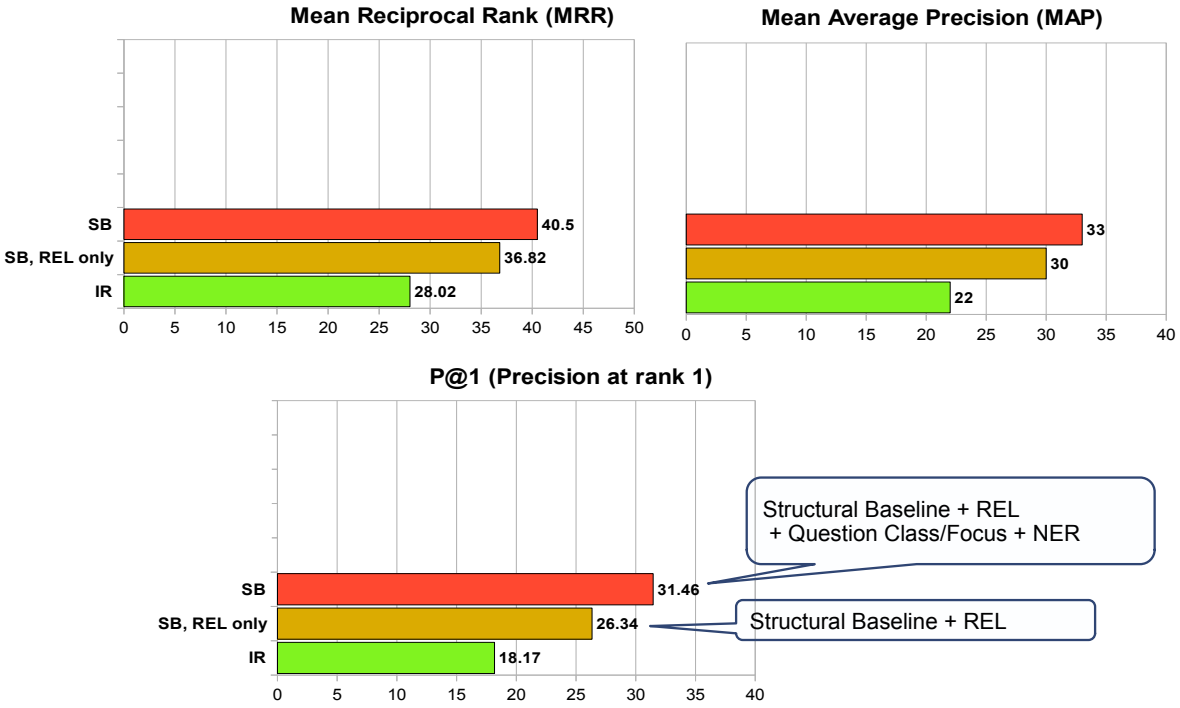
# Baselines

- IR baseline
  - Terrier engine, **BM25** model
- Structural baseline (Severyn&Moschitti, CoNLL '13)

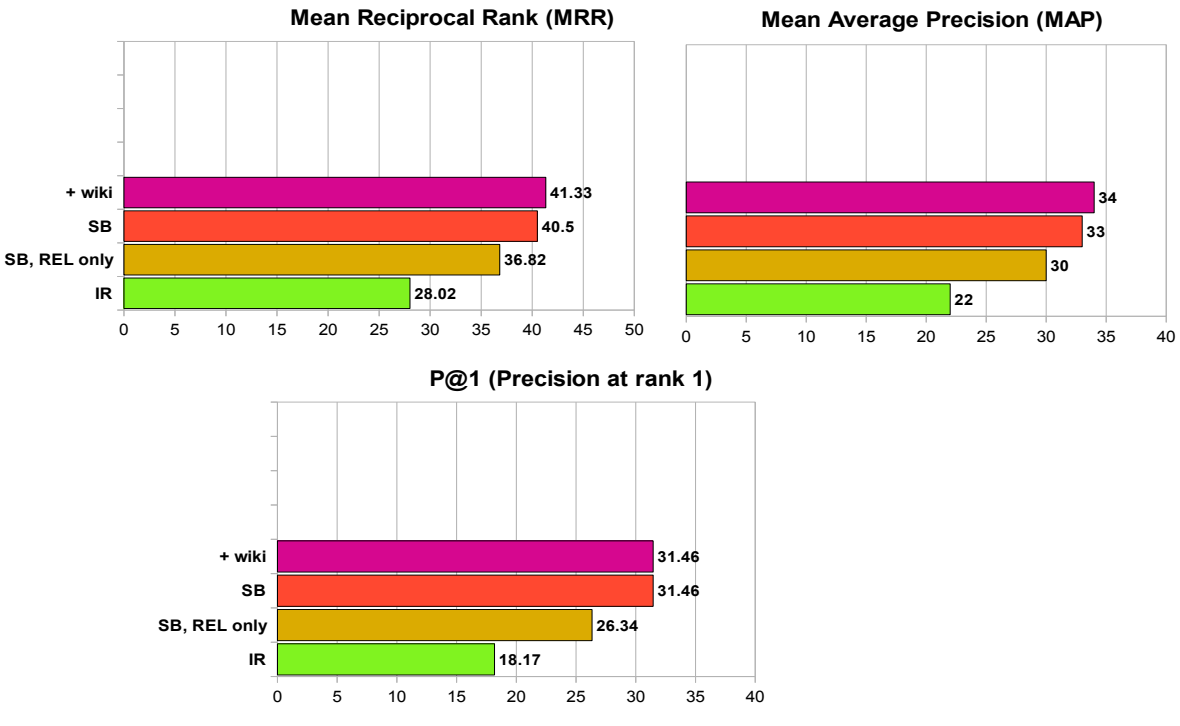
# Baselines

- IR baseline
  - Terrier engine, **BM25** model
- Structural baseline (Severyn&Moschitti, CoNLL '13)
  - **V**: feature vector
    - Question (Q) /Answer Passage (AP) **cosine BOW** similarity
    - Q/AP Partial Tree Kernel (**PTK**) similarity
    - normalized **IR BM25** score

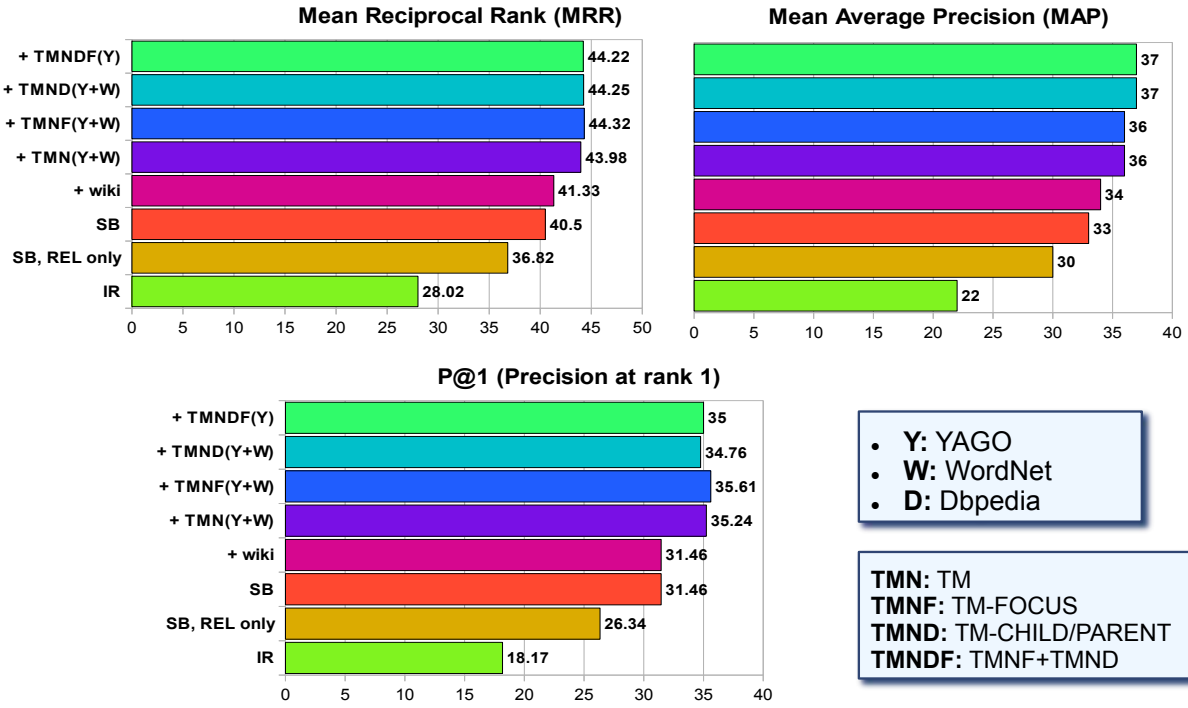
# Results




# Results



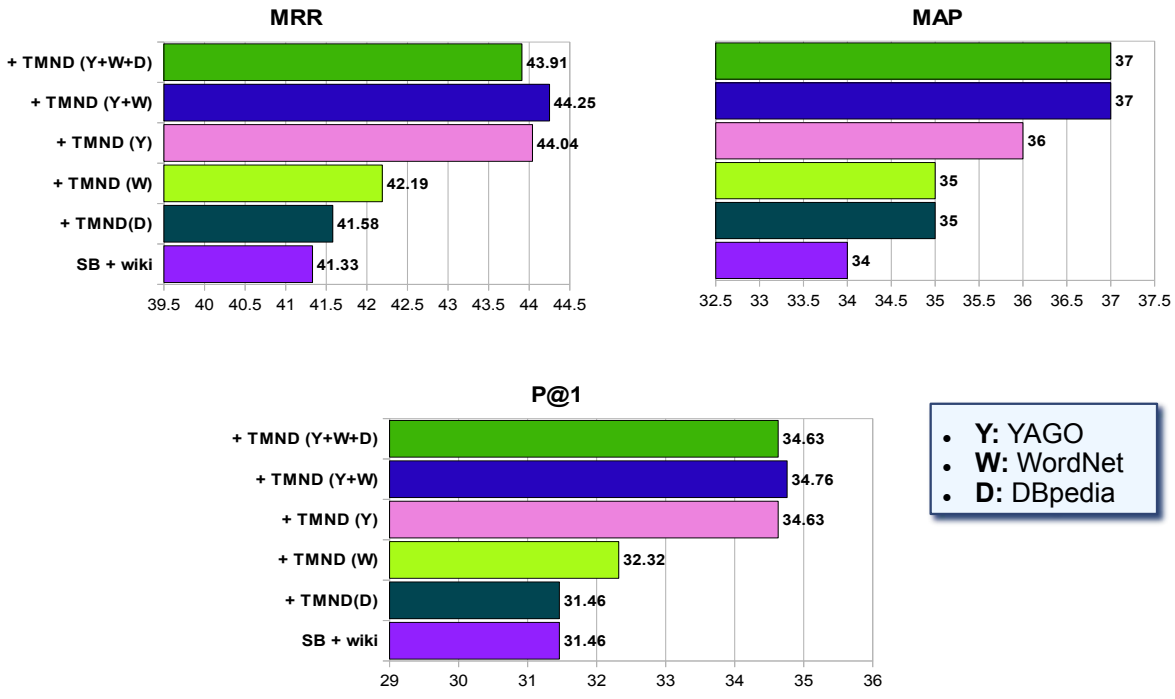
# Results




- **Y:** YAGO
- **W:** WordNet
- **D:** Dbpedia

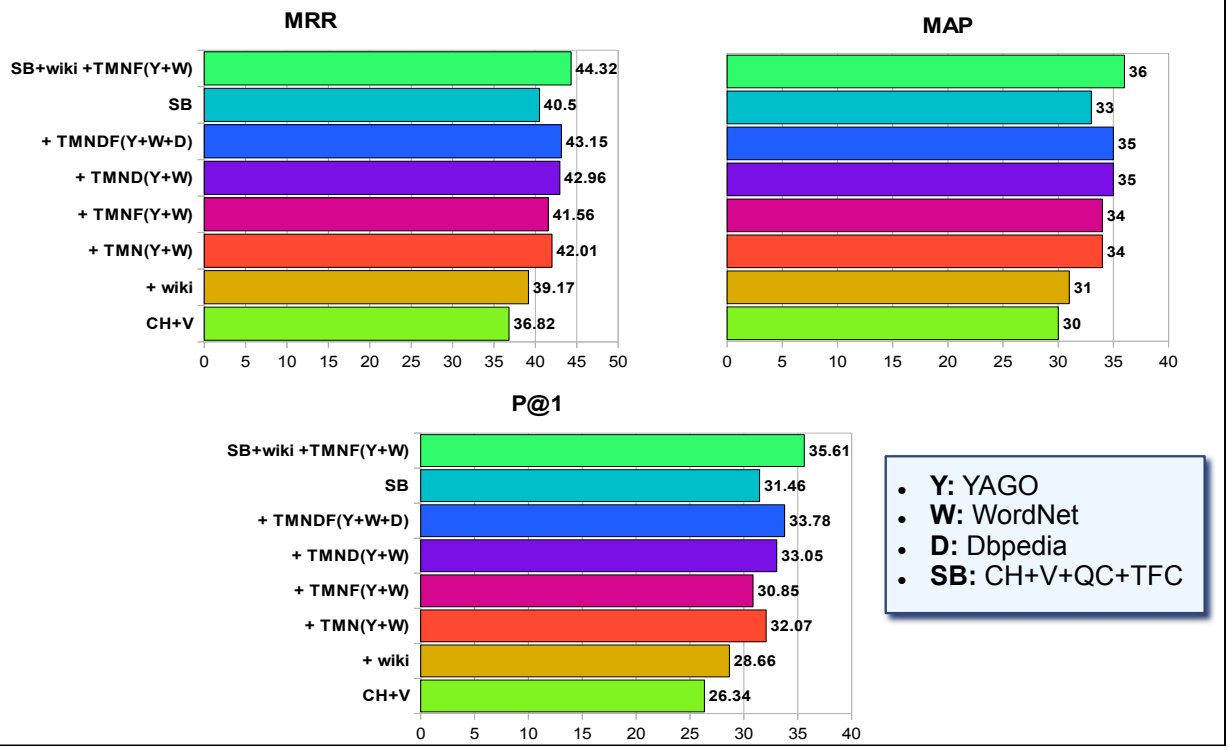
**TMN:** TM
**TMNF:** TM-FOCUS
**TMND:** TM-CHILD/PARENT
**TMNDF:** TMNF+TMND

# Results: Data sources impact




- **Y:** YAGO
- **W:** WordNet
- **D:** DBpedia

# Results: CH+V+TM




# State-of-art approaches

- Feature-based models based on
  - Quasi-synchronous grammars (Wang, 2007)
  - Tree Edit Distance (Heilman & Smith, 2010)
  - Probabilistic model to learn TED transformations on dependency trees (Wang & Manning, 2010)
  - CRF + TED features (Yao et al., 2013)
- Structural representation based approaches
  - SVM + shallow parse tree representation (Severyn & Moschitti, 2012), (Severyn et al, 2013)

Our baseline

# TREC'13 academic benchmark

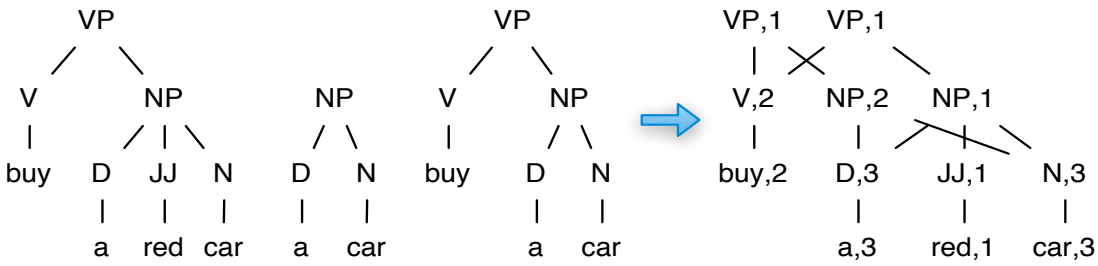
- Factoid open-domain TREC QA corpus prepared by Wang et al. (2007)
- Training data from the 1,229 TREC8-12 questions
  - Training questions automatically marked using regular expressions
  - The test data contains 89 questions, whose answers were manually annotated
- We used 10 answer passages for each question for training and all the passages for testing
  - passages are given (no search engine is needed)

# Latest Results on TREC'13

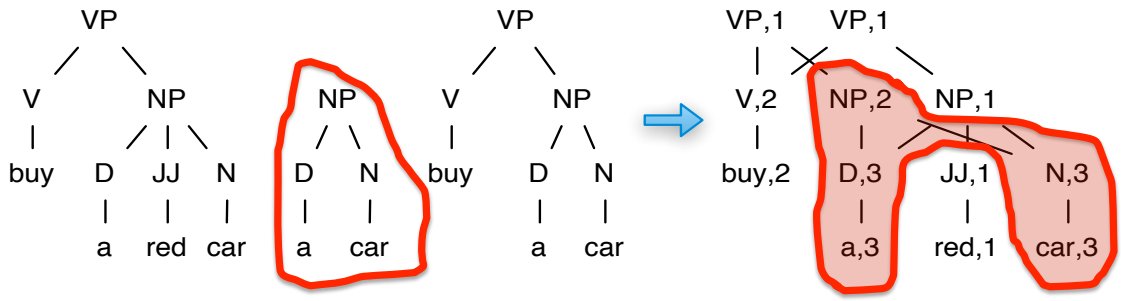
| | Map | MRR |
|---|---|---|
| Yao et al., 2013 [35] | 63.07 | 74.77 |
| CH+V | 65.66 | 74.59 |
| DEP+V | 65.87 | 72.68 |
| CH+V+QC+TFC | 67.55 | 75.14 |
| CH+V+QC+TFC* (SB) | 67.42 | 75.06 |
| DEP+V+QC+TFC | 65.78 | 70.79 |
| $SB_w$ | 69.49 | 74.73 |
| $+TM_N$:Y+W+D | 70.75 | 77.71 |
| $+TM_{NF}$ :Y+W+D | $71.03^{\dagger}$ | **78.03** |
| $+TM_{ND}$:Y+W+D | $71.60^{\ddagger}$ | 78.60 |
| $+TM_{NDF}$:Y+W+D | $\mathbf{71.31}^{\dagger}$ | 77.74 |
| CH+V+QC+TFC+SRL | 67.91 | 75.66 |

# A glimpse to the exploitation of Direct Acyclic Graph (DAG)

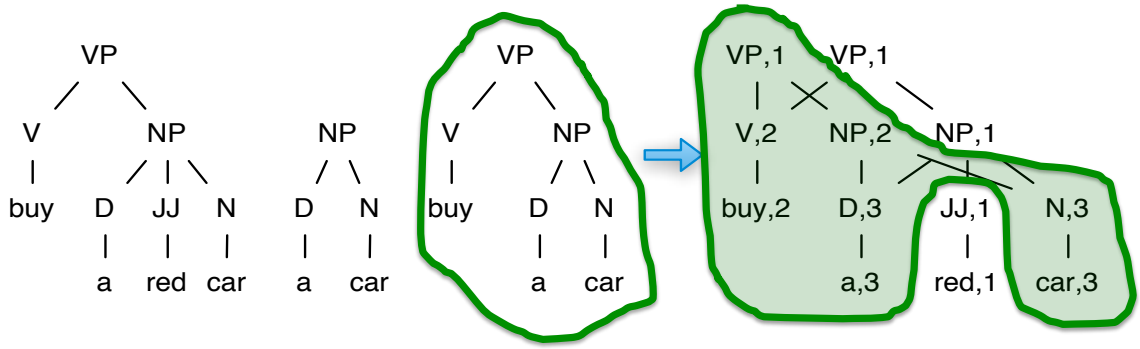
# Three syntactic trees and the resulting DAG



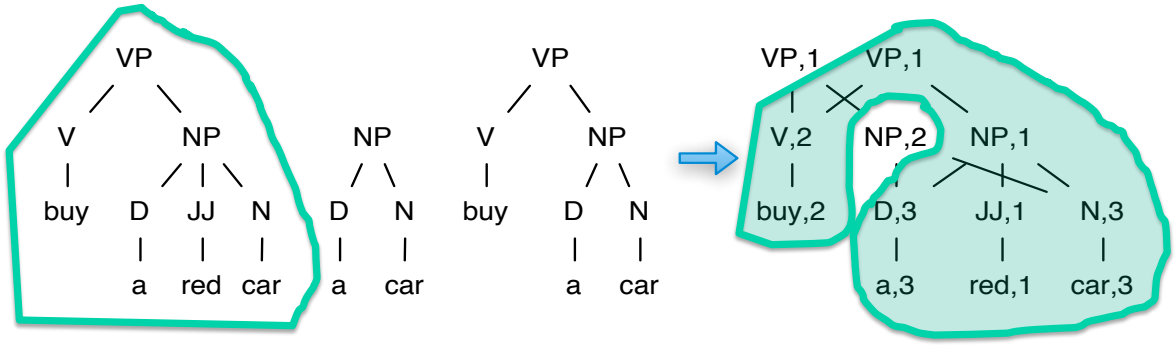
# Three syntactic trees and the resulting DAG




# Three syntactic trees and the resulting DAG



# Three syntactic trees and the resulting DAG



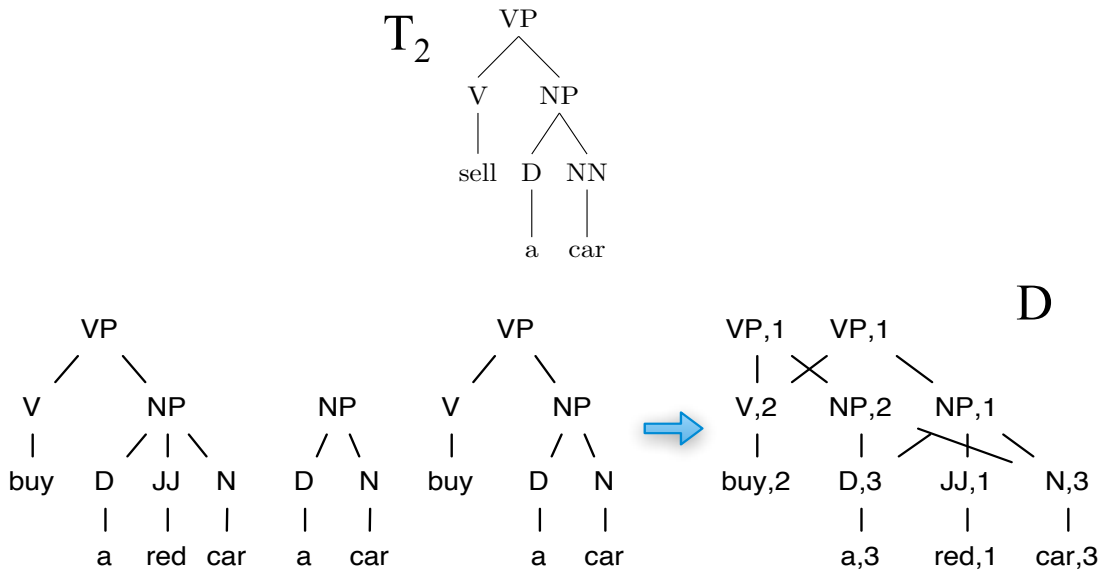

# DAG Kernel
## (Severyn&Moschitti, ECML2011)

**Theorem 1.** *Let $D$ be a DAG representing a tree forest $F$ and $K_{dag}(D, T_2) = \sum_{n_1 \in N_D} \sum_{n_2 \in N_{T_2}} f(n_1)\Delta(n_1, n_2)$ then*

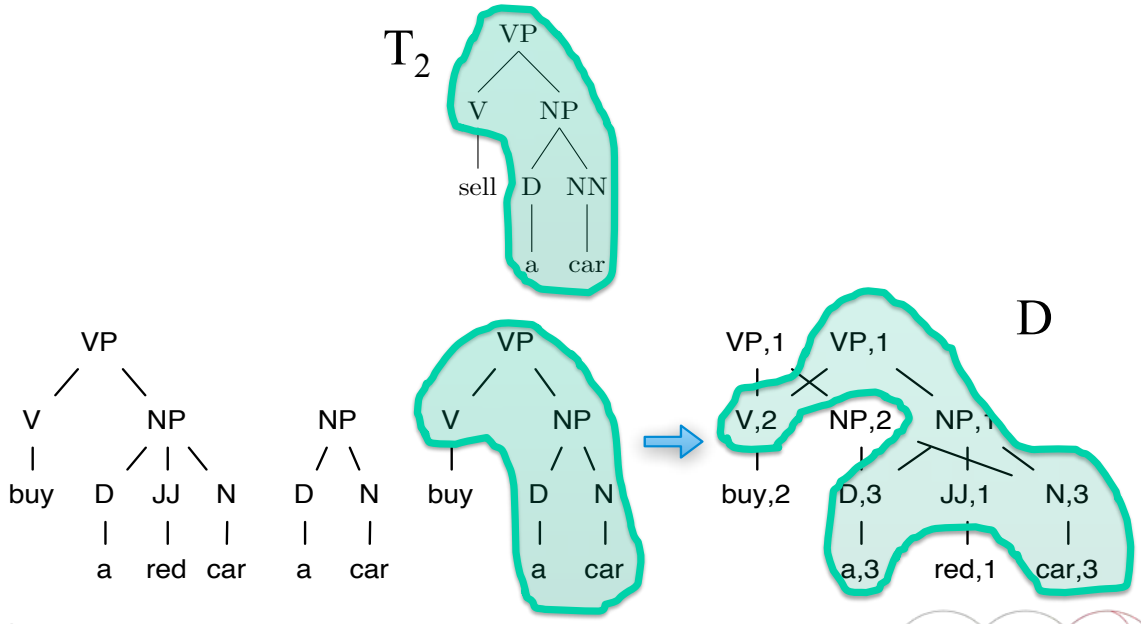
$$\sum_{T1 \in F} TK(T_1, T_2) = K_{dag}(D, T_2), \tag{4}$$

*where $f(n_1)$ is the frequency associated with $n_1$ in the DAG.*

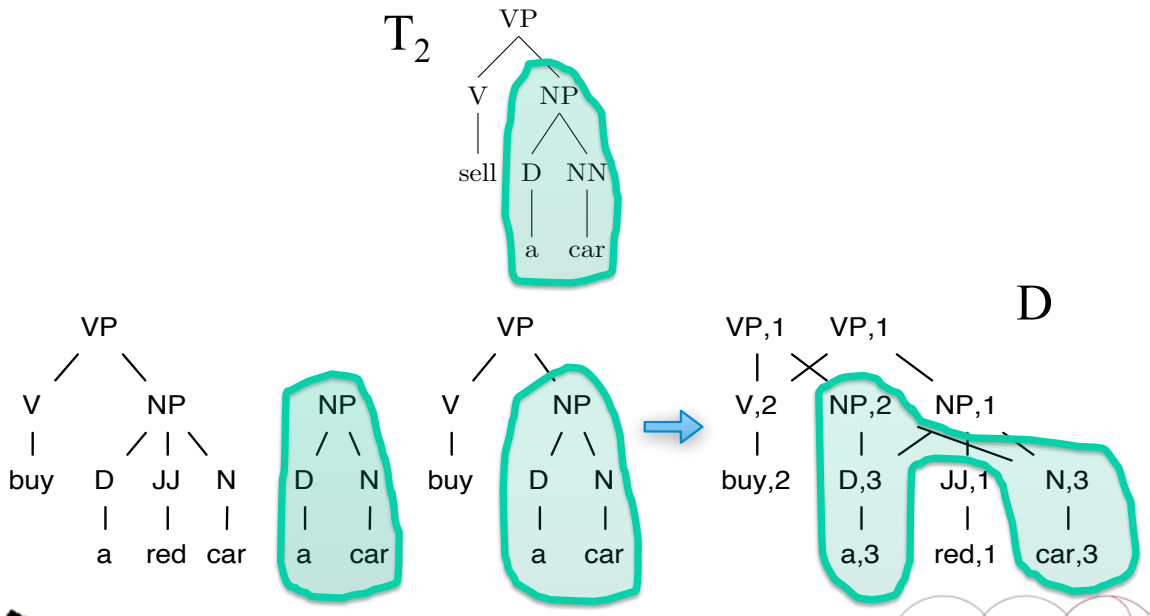
# Computation example




# Computation example



## Computation example




## Conclusions

- Relational Learning from pairs of texts offers great potential
  - many applications, ranging from QA to MT
- Using semantic and structural representations is difficult:
  - How to engineer rules for exploiting syntactic/semantic information?
  - How to engineer features for learning algorithms?
- We can use powerful ML algorithms and kernel methods
  - Kernels can generate many features
  - SVMs are robust to noise and irrelevant features
- State of the art in QA and other relational learning tasks

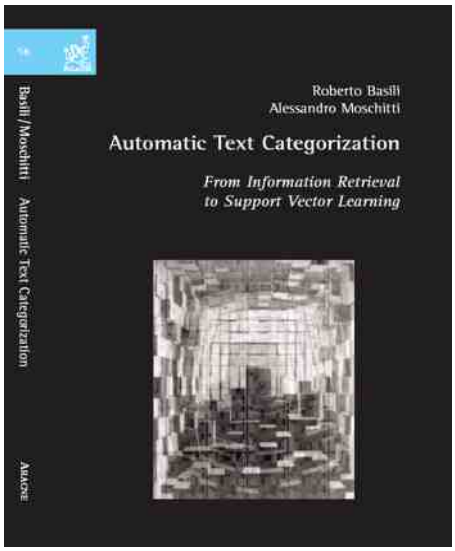
## Future (on going work)

- Deeper modeling of paragraphs: *shallow semantics and discourse structures* to design more compact and accurate representation of whole paragraphs
- Applying automatic JHU-PIRE MR
- Use of reverse kernel engineering to build efficient systems: [Pighin&Moschitti, CoNLL2009, EMNLP2009, CoNLL2010]

## Documentation

- Tutorial Webpage
  - http://disi.unitn.it/moschitti/SIGIR-tutorial.htm
  - Software
  - Data: Question Classification and Paragraph reranking
  - Updated slides
  - Papers
  - Books

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




# Forthcoming 2014

- *State-of-the-art Kernels in Natural Language Processing*
  Author: Alessandro Moschitti
  Synthesis Lectures on Human Language Technologies
  Editor: Morgan & Claypool Publishers

# Thank you

# References


- Qi Ju, Alessandro Moschitti: *Incremental Reranking for Hierarchical Text Classification*. ECIR 2013: 726-729
- Qi Ju, Alessandro Moschitti, Richard Johansson: *Learning to Rank from Structures in Hierarchical Text Classification*. ECIR 2013: 183-194
- Aliaksei Severyn and Massimo Nicosia and Alessandro Moschitti. *iKernels-Core: Tree Kernel Learning for Textual Similarity*. In *SEM 2013.
- Aliaksei Severyn and Alessandro Moschitti. *Fast Linearization of Tree Kernels over Large-Scale Data*. In IJCAI 2013.
- Aliaksei Severyn and Massimo Nicosia and Alessandro Moschitti. *Building Structures from Classifiers for Passage Reranking*. To appear in CIKM 2013.
- Barbara Plank and Alessandro Moschitti. *Embedding Semantic Similarity in Tree Kernels for Domain Adaptation of Relation Extraction*. To appear in ACL 2013.

# References


- Aliaksei Severyn and Massimo Nicosia and Alessandro Moschitti. *Learning Adaptable Patterns for Passage Reranking*. In CoNLL 2013.
- Aliaksei Severyn and Massimo Nicosia and Alessandro Moschitti. *Learning Semantic Textual Similarity with Structural Representations.* In ACL 2013
- Aliaksei Severyn, Alessandro Moschitti: *Structural relationships for large-scale learning of answer re-ranking*. SIGIR 2012: 741-750
- Alessandra Giordani, Alessandro Moschitti, *Translating Questions to SQL Queries with Generative Parsers Discriminatively Reranked*. Proceedings of the 24rd International Conference on Computational Linguistics (Coling), Bombai, Mumbai, India, 2012.
- Alessandro Moschitti, Qi Ju, Richard Johansson: *Modeling Topic Dependencies in Hierarchical Text Categorization*. ACL 2012: 759-767


# References


- Aliaksei Severyn, Alessandro Moschitti: *Fast Support Vector Machines for Convolution Tree Kernels*. Data Mining Knowledge Discovery 25: 325-357, 2012.
- Siddharth Patwardhan, Branimir Boguraev, Apoorv Agarwal, Alessandro Moschitti, Jennifer Chu-Carroll, *Labeling by Landscaping: Classifying Tokens in Context by Pruning and Decorating Trees*. In Proceedings of the Conference on Information and Knowledge Management, Maui Hawaii, CIKM 2012.
- Danilo Croce, Alessandro Moschitti, Roberto Basili, Martha Palmer: *Verb Classification using Distributional Similarity in Syntactic and Semantic Structures*. ACL 2012: 263-272
- Vien Nguyen and Alessandro Moschitti, *Structural Reranking Models for Named Entity Recognition*, Special issue on Natural Language Processing in the Web Era, Intelligenza Artificiale, IOS Press, Volume 6.2, 2012.


# References


- M. Dinarelli, A. Moschitti, and G. Riccardi. *Discriminative Reranking for Spoken Language Understanding*. IEEE Transaction on Audio, Speech and Language Processing, 2012.
- Alessandro Moschitti and Silvia Quarteroni, *Linguistic Kernels for Answer Re-ranking in Question Answering Systems,* Information and Processing Management: an International journal, ELSEVIER, 2011
- Danilo Croce, Alessandro Moschitti, and Roberto Basili. *Structured lexical similarity via convolution kernels on dependency trees*. In Proceedings of EMNLP, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.
- Aliaksei Severyn and Alessandro Moschitti. *Fast support vector machines for structural kernels*. In ECML-PKDD, 2011, Greece, 2011. Best Machine Learning Student Paper Award


# References


- Truc Vien T. Nguyen and Alessandro Moschitti. *Joint distant and direct supervision for relation extraction*. In Proceedings of the The 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand, November 2011, Association for Computational Linguistics.
- Alessandro Moschitti, Jennifer Chu-carroll, Siddharth Patwardhan, James Fan, and Giuseppe Riccardi. *Using syntactic and semantic structural kernels for classifying definition questions in Jeopardy!* In Proceedings of EMNLP, pages 712–724, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.
- Truc Vien T. Nguyen and Alessandro Moschitti. *End-to-end relation extraction using distant supervision from external semantic repositories*. In Proceedings of HLT-ACL, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
- *Large-Scale Support Vector Learning with Structural Kernels*, In Proceedings of the 21th European Conference on Machine Learning (ECML-PKDD2010), Barcelona, Spain, 2010.


# References


- Alessandro Moschitti' handouts http://disi.unitn.eu/moschitti/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.


# 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.