Recognizing Textual Entailment Using a Subsequence Kernel Method

Rui Wang & Günter Neumann

LT Lab at DFKI

Saarbrücken, Germany

Recognizing Textual Entailment (RTE)

- ☆ Motivation: textual variability of semantic expression
- ☆ Idea: given two text expressions T & H:
  - Does text T justify an inference to hypothesis H?
  - Is H semantically entailed in T ?

Edward VIII shocked the world in 1936 when he gave up his throne to marry an American divorcee, Wallis Simpson.

?

King Edward VIII abdicated in 1936.

- ☆ PASCAL Recognising Textual Entailment Challenge
  - since 2005, cf. Dagan et al.
  - 2007: 3rd RTE challenge, 25 research groups participated
- ☆ A core technology for text understanding applications:
  - Question Answering, Information Extraction, Semantic Search, Document Summarization, …

Processing of real text documents

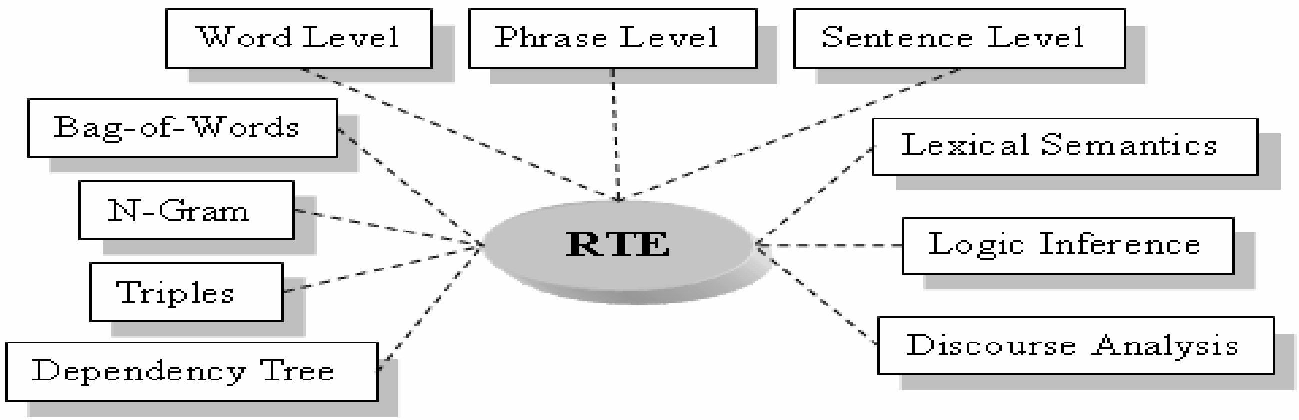
☆ Semantic under-specification

- Imprecise expressed semantic relationships
- Vagueness, ambiguity

☆ Error tolerant methods needed

- Noisy input data
- Noisy intermediate component output

Different approaches consider/integrate features from different linguistics levels



Our goal: How far can we get with syntax only ?

☆ Subtree alignment on syntactic level

- Check similarity between tree of H and relevant subtree in T

☆ Tree compression (redundancy reduction)

- Reduce noise from input/parsing
- Yields compressed path-root-path sequences

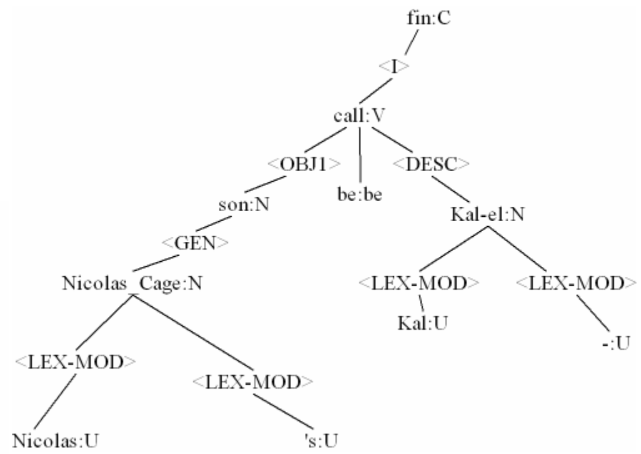
☆ Subsequence kernel

- Consider all possible subsequence of spine (path) difference pairs
- SVM for classification

Sentence representation

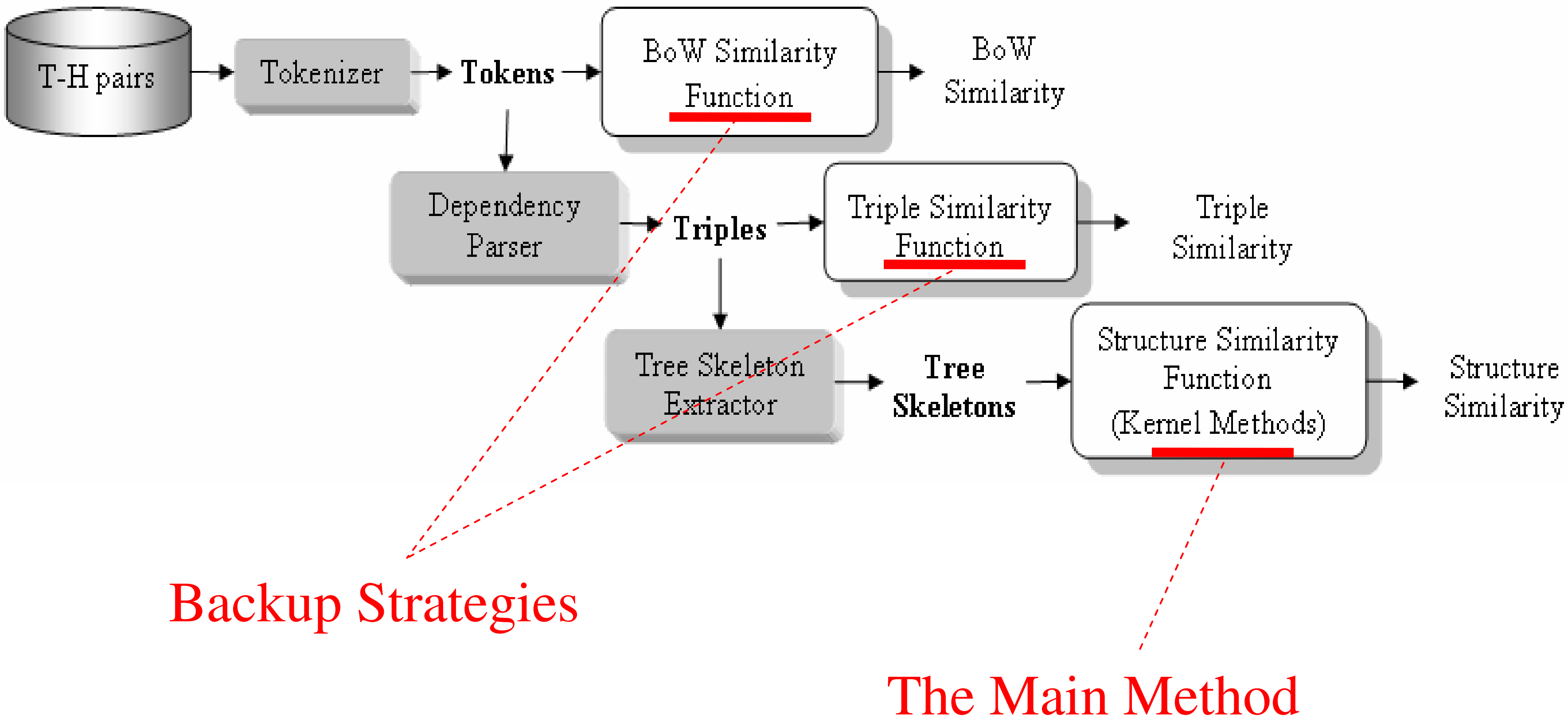
- ☆ A sentence is represented as a set of triples of general form <head relation modifier>
  - – Ex: Nicolas Cage’s son is called Kal’el

```
<triple left="E0" right="6">fin:C i call:V</triple>
<triple left="2" right="1">Nicolas_Cage:N lex-mod Nicolas:U</triple>
<triple left="2" right="3">Nicolas_Cage:N poss 's:U</triple>
<triple left="4" right="2">son:N gen Nicolas_Cage:N</triple>
<triple left="6" right="4">call:V s son:N</triple>
......
```

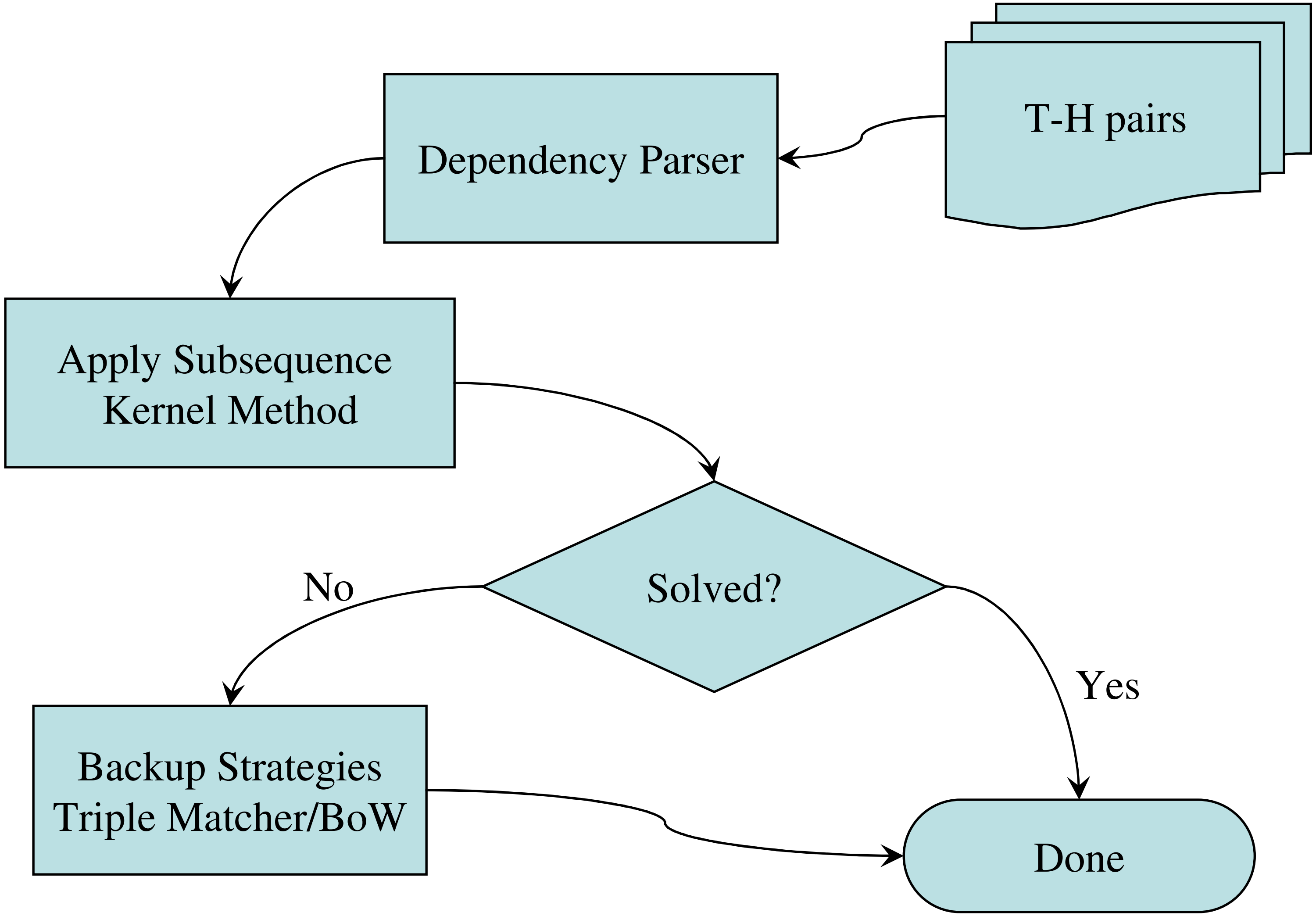


- ☆ Dependency Structure
  - – A DAG where nodes represent words and edges represent directed grammatical functions
  - – We consider this as a “shallow semantic representation”
  - – We use Minipar (Lin, 1998) and StanfordParser (Klein and Manning, 2003) as current parsing engines

System Overview: Feature Extraction

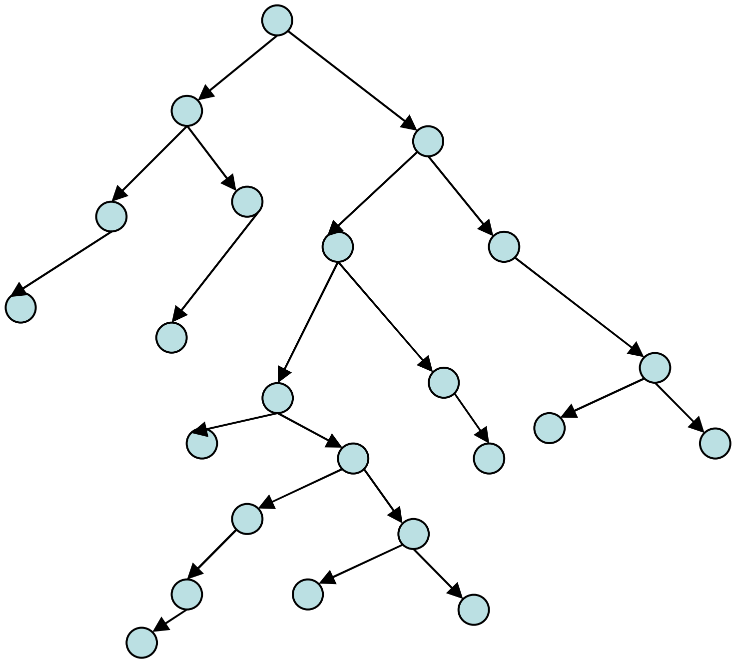


System Workflow

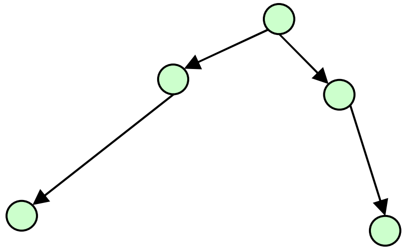


Basic idea, step 1: Dependency parsing

Dependency Tree for T



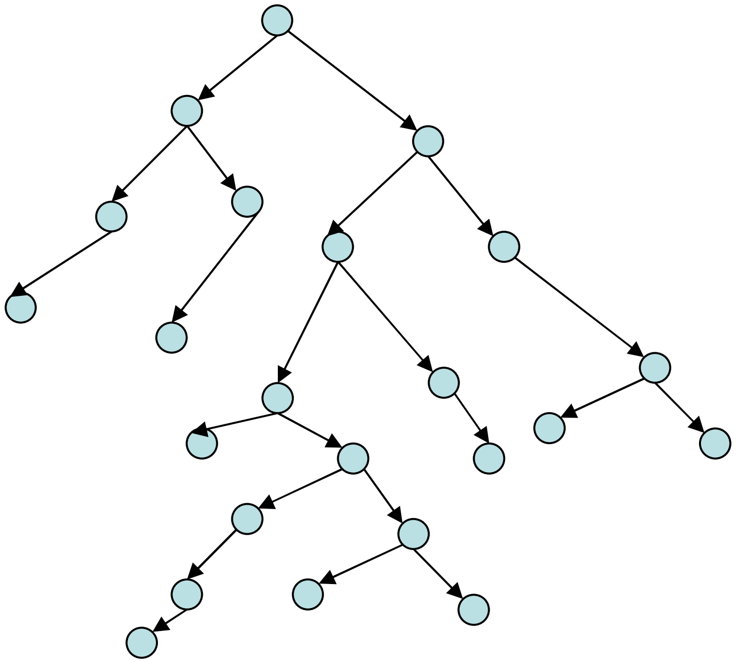
Dependency Tree for H

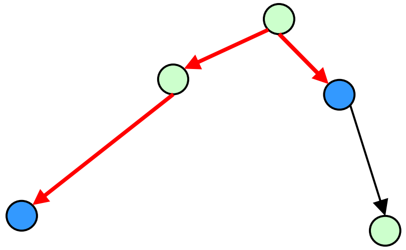


Basic idea, step 2: verb/noun subtree of H

Dependency Tree for T

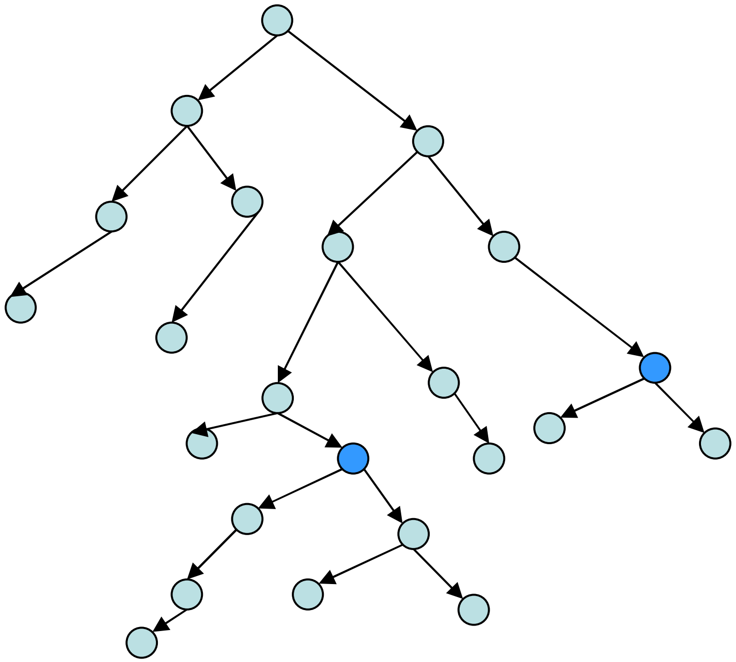
Dependency Tree for H



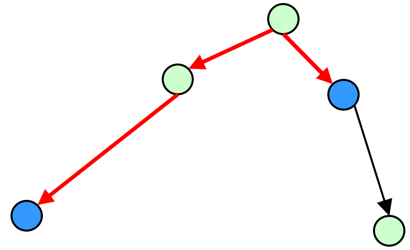


Basic idea, step 3: Foot node alignment

Dependency Tree for T

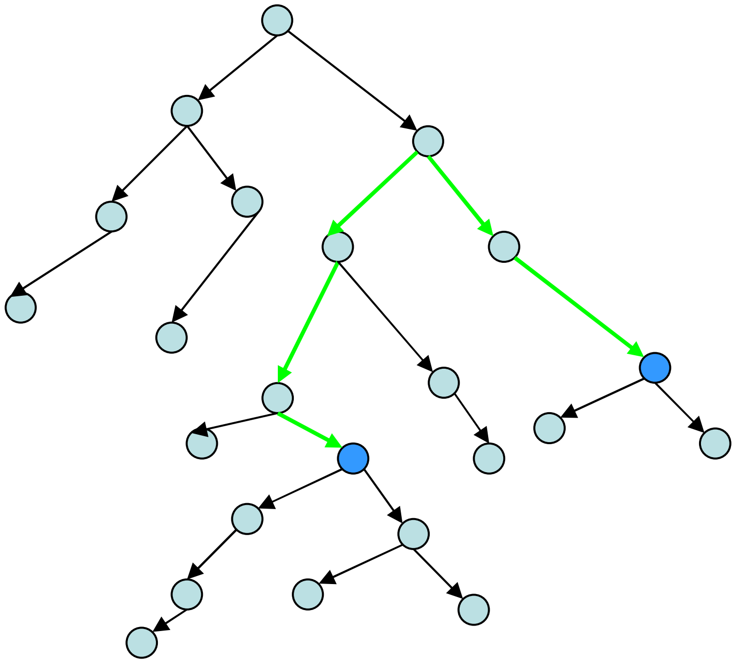


Dependency Tree for H

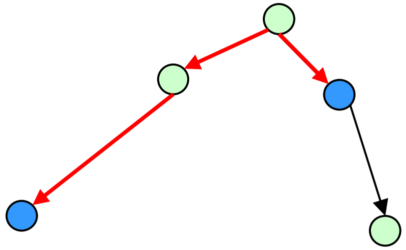


Basic idea, step 4: Root node identification in T

Dependency Tree for T

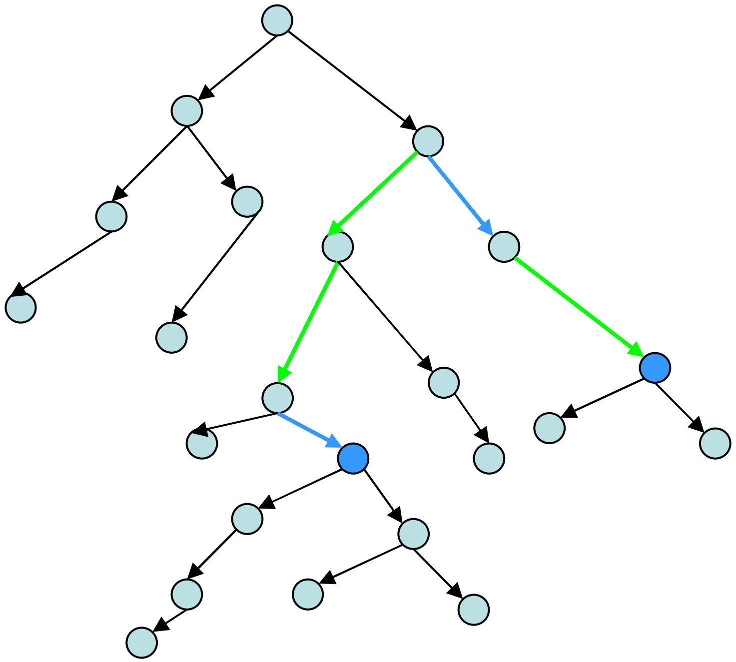


Dependency Tree for H

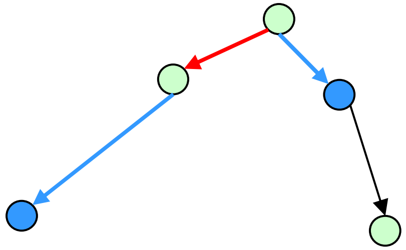


Basic idea, step 5: Spine Difference

Dependency Tree for T

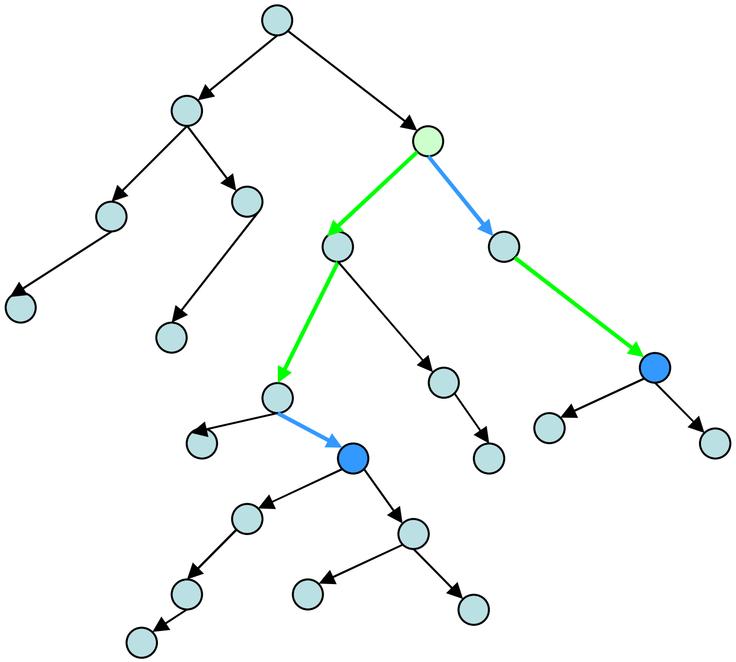


Dependency Tree for H

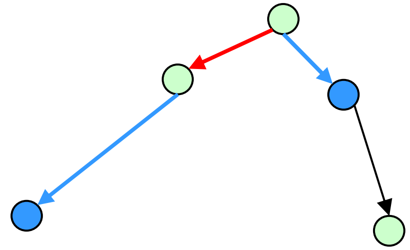


Basic idea, step 6: Root node alignment

Dependency Tree for T

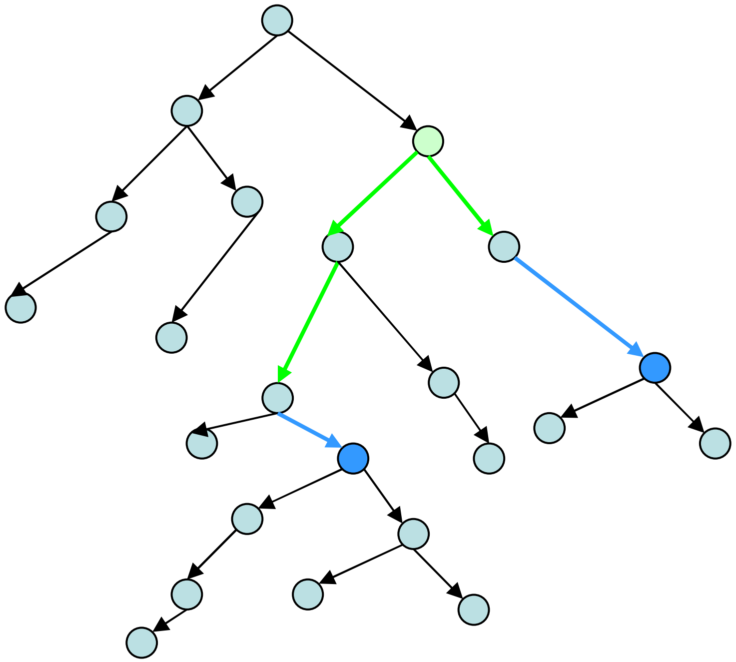


Dependency Tree for H

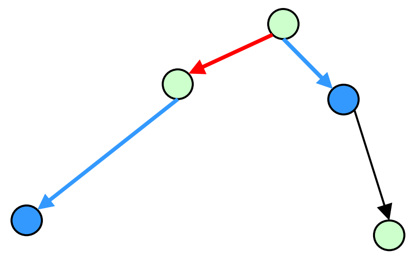


Basic idea, step 7: Feature extraction

Dependency Tree for T

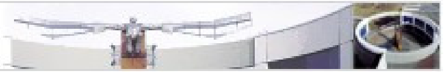


Dependency Tree for H



Elementary Predicate	Left spine diff.	Right spine diff.	Verb cons.
T:			1
H:		ε	

A Natural Language Example



☆ Pair: id=*“61"* entailment=*“**YES**“* task=*“**IE**“* source=*“RTE”*

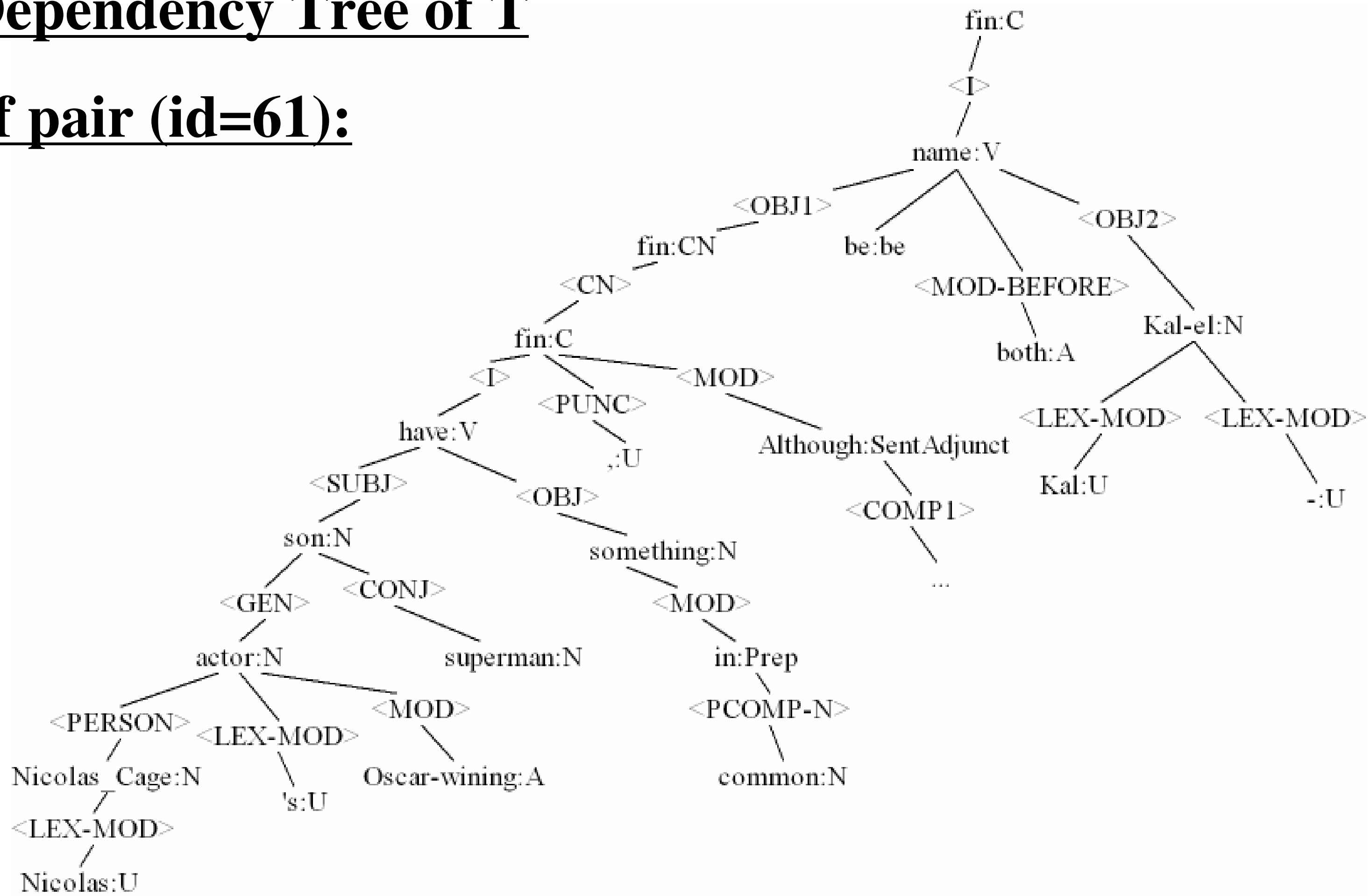
– Text:

*Although they were born on different planets, Oscar-winning actor **Nicolas Cage**'s new **son** and Superman have something in common, both were named **Kal-el**.*

– Hypothesis:

***Nicolas Cage**'s **son** is called **Kal-el**.*

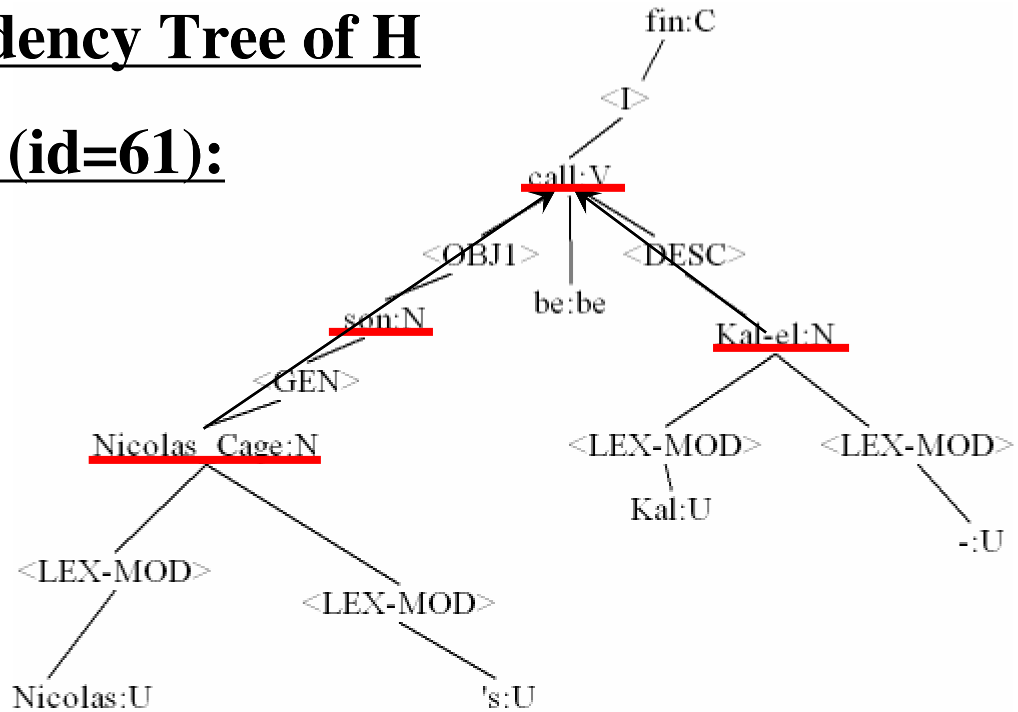
Dependency Tree of T of pair (id=61):



Dependency Graph (cont.)

Dependency Tree of H

of pair (id=61):

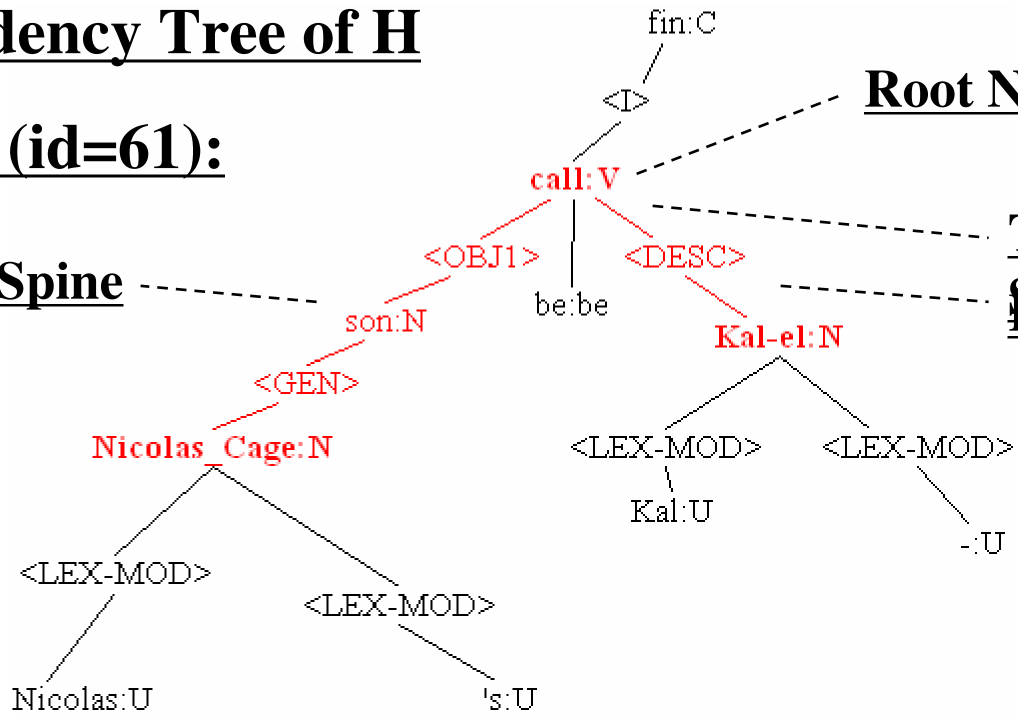


- **Observations**
  - **H is simpler than T**
  - **H can help us to identify the relevant parts in T**

Nicolas Cage's son *is called* ***Kal-el.***

Tree Skeleton

Dependency Tree of H of pair (id=61):



Root Node

Tree Skeleton:

Left Spine

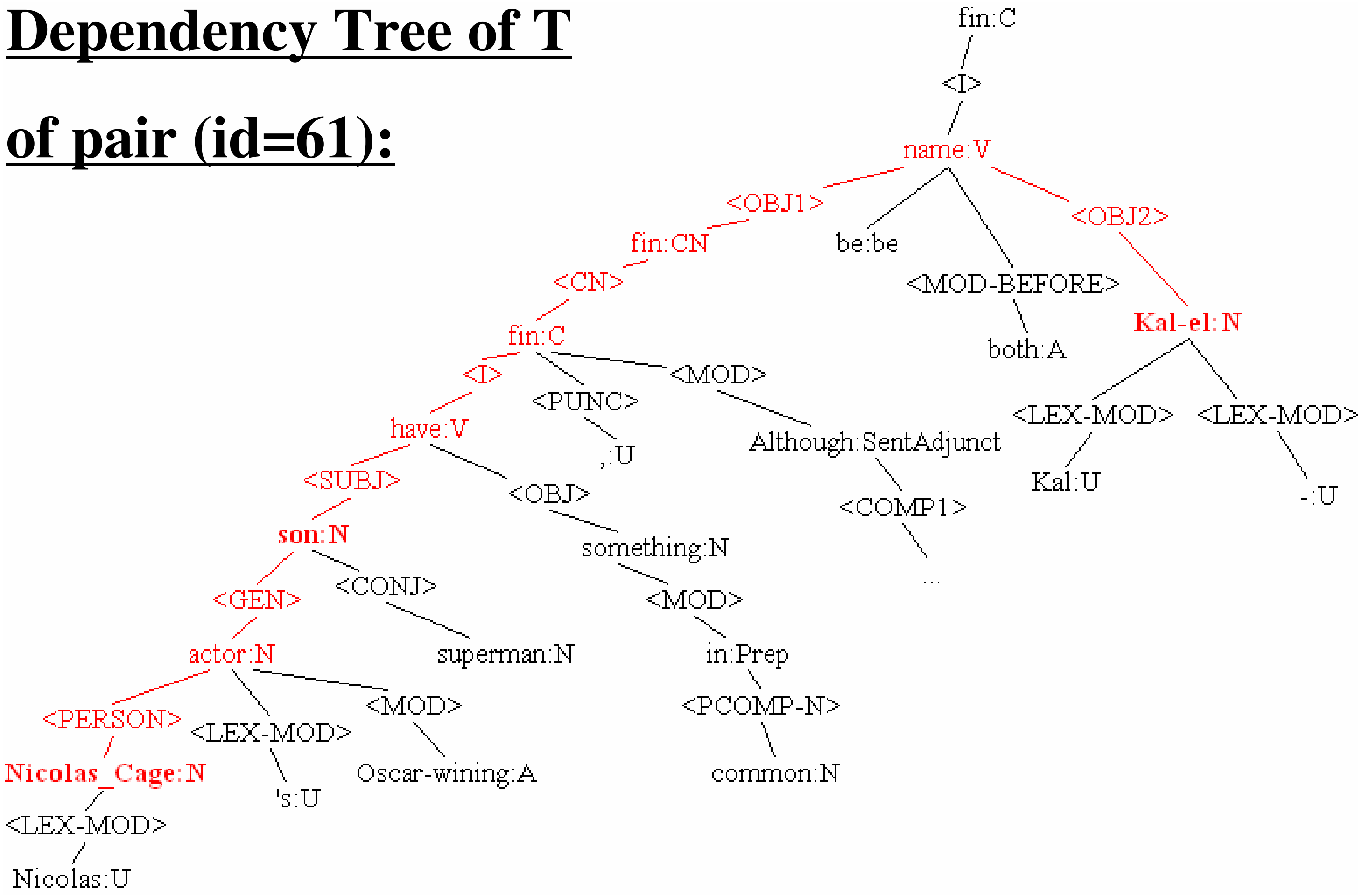
Right Spine

Nicolas Cage*'s* ***son*** *is called* ***Kal-el****.*

Tree Skeleton (cont.)

Dependency Tree of T

of pair (id=61):



☆ Left Spine #Root Node# Right Spine

– Text

Nicolas_Cage:N <PERSON> actor:N <GEN> son:N <SUBJ> have:V <I> fin:C <CN> fin:CN <OBJ1>
#Name:V#
<OBJ2> Kal-el:N

↓

Nicolas_Cage:N & N <GEN> son:N <SUBJ> V <I> C <CN> CN <OBJ1> #Name:V# <OBJ2> Kal-el:N

↓

Nicolas_Cage:N <GEN> son:N <SUBJ> V <SUBJ> #name:V# <OBJ> Kal-el:N

– Hypothesis

Nicolas_Cage:N <GEN> son:N <SUBJ> #call:V# <OBJ> Kal-el:N

- ☆ Merging
  - – Left Spines: exclude Longest Common Prefixes
  - – Right Spines: exclude Longest Common Suffixes
- ☆ RootNode Comparison
  - – Verb Consistence (VC)
  - – Verb Relation Consistence (VRC)

Left Spine Difference (LSD)

Nicolas_Cage:N <GEN> son:N <SUBJ> V <SUBJ> #name:V# <OBJ> Kal-el:N

Nicolas_Cage:N <GEN> son:N <SUBJ> #call:V# <OBJ> Kal-el:N

☆ Pattern Format

- <LSD, RSD, VC, VRC> → Predication
- Example: *<"SUBJ V", "", 1, 1>* → *YES*

☆ Closed-Class Symbol (CCS)

Types	Symbols
Dependency Relation Tags	*SUBJ, OBJ, GEN, …*
POS Tags	*N, V, Prep, …*

- LSD and RSD are either *NULL* or CCS sequences

Testing Phase

☆ Pair: id="*247*" entailment="*YES*" task="*IE*" source="*BinRel*"

– Text:

Author Jim Moore was invited to argue his viewpoint that Oswald , acting alone , killed Kennedy.

– Hypothesis:

Oswald killed Kennedy.

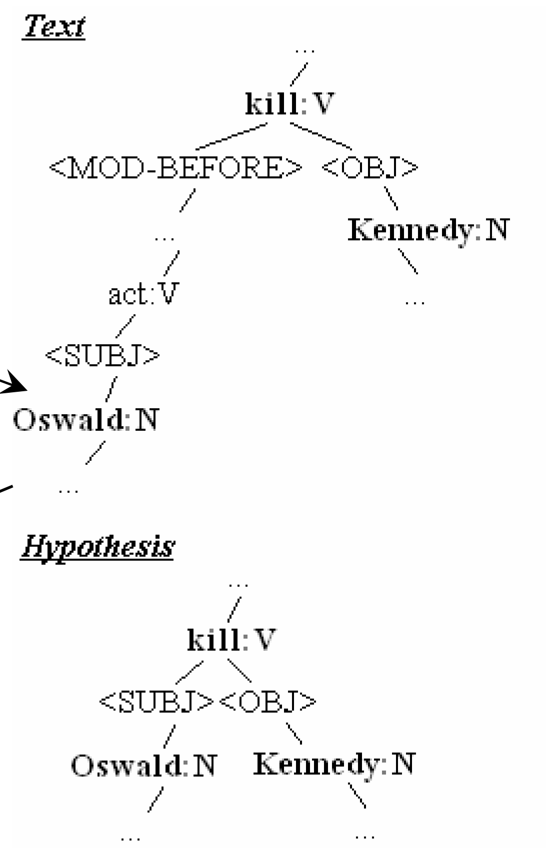
Testing Phase (cont.)

Text

```
......
<triple left="17" right="E0">kill:V mod-before vpsc:C</triple>
<triple left="17" right="16">kill:V punc ,:U</triple>
<triple left="17" right="E8">kill:V subj Oswald:N</triple>
<triple left="17" right="18">kill:V obj Kennedy:N</triple>
......
```

Hypothesis

```
<triple left="E0" right="2">fin:C i kill:V</triple>
<triple left="2" right="1">kill:V s Oswald:N</triple>
<triple left="2" right="E2">kill:V subj Oswald:N</triple>
......
```



Oswald:N <u><SUBJ> V</u> <SUBJ> #kill:V# <OBJ> Kennedy:N

Oswald:N <SUBJ> #kill:V# <OBJ> Kennedy:N

<u><"SUBJ V", "", 1, 1> → YES</u>

☆ Entailment methods:

- Bag-of-Words (BoW)
- Triple Set Matcher (TSM)
- Minipar + Sequence Kernel + Backup Strategies (Mi+SK+BS)
- StanfordParser + Sequence Kernel + Backup Strategies (SP+SK+BS)

☆ Classifier:

- SVM (SMO) classifier from the WEKA ML toolkit

Experiments: Data

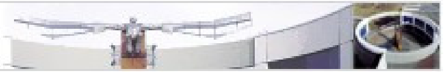
☆ From RTE challenges:

- RTE-2 Dev Set (800 **T**-**H** pairs) + Test Set (800 **T**-**H** pairs)
- RTE-3 Dev Set (800 **T**-**H** pairs) + Test Set (800 **T**-**H** pairs)

☆ Additional data for IE and QA tasks:

- Automatically collected from MUC6, BinRel (*Roth and Yih, 2004*), TREC-2003
- Manually classified into yes/no concerning entailment relation

Results on RTE-2 Data



Systems\Tasks	IE	IR	QA	SUM	ALL
Exp A1: 10-Fold Cross-Validation on Dev+Test Set					
BoW	**50%***	58.8%	58.8%	74%	**60.4%**
TSM	**50.8%**	57%	62%	70.8%	**60.2%**
Mi+SK+BS	**61.2%**	58.8%	63.8%	74%	**64.5%**
Exp A2: Train: Dev Set (50%); Test: Test Set (50%)					
BoW	**50%**	56%	60%	66.5%	**58.1%**
TSM	**50%**	53%	64.5%	65%	**58.1%**
Mi+SK+BS	**62%**	61.5%	64.5%	66.5%	**63.6%**

* The accuracy is actually 47.6%. Since random guess will achieve 50%, we take this for comparison.

Results on RTE-3 Data

Systems\Tasks	IE	IR	QA	SUM	All
Exp B1: 10-fold Cross Validation on RTE-3 Dev Data					
BoW	54.5%	70%	76.5%	**68.5%**	67.4%
TSM	53.5%	60%	68%	62.5%	61.0%
Mi+SK+BS	**63%**	74%	79%	68.5%	**71.1%**
SP+SK+BS	**60.5%**	70%	**81.5%**	68.5%	**70.1%**
Exp B2: Train: Dev Data; Test: Test Data					
Mi+SP+SK+BS	58.5%	70.5%	79.5%	59%	**66.9%***

* The 5th place of RTE-3 among 26 teams

Components of the 5th best systems

Systems	Acc. %	Lx*	Ng	Sy	Se	LI	C	ML	B
Hickl et al.	80,00	X	X	X	X		X	X	X
Tatu et al.	72,25	X				X			X
Iftene	69,13	X		X					X
Adams	67,00	X	X				X	X	
DFKI	66,87			X				X	

* Following the notation in (*Giampiccolo et al., 2007*):

Lx: Lexical Relation DB;

Ng: N-Gram / Subsequence overlap;

Sy: Syntactic Matching / Alignment;

Se: Semantic Role Labeling;

LI: Logical Inference;

C: Corpus/Web;

ML: ML Classification;

B: Entailment corpora/Background Knowledge;

Comments

☆ Puristic approach:

– We do not exploit any additional knowledge source beside the dependency trees nor have we extended the RTE training data

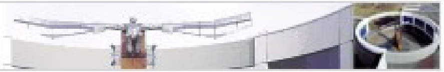
☆ Relational method:

– For the **IE** task, SK method gives highest improvements

– Kernel method seem to be more appropriate if the underlying task reveals a more "relational nature"

☆ Fallback strategies:

– The "shallow" methods realized through BoW and TSM seem to work better for IR and SUM.



☆ **IE:** MUC6, BinRel Corpus

- **T**: relevant sentence(s)
- **H**: NE + Relation + NE

Dole had hoped to pull out a win in North Carolina, the home state of his wife, Elizabeth.

T

Elizabeth is born in North Carolina.

☆ **QA:** TREC2003 QA

- **T**: (ir)relevant sentence(s)
- **H**: question + answer

Vice-President Albert Gore described the book "critically important" and compared it with "Silent Spring," Rachel Carson's 1962 book that set off a movement to ban DDT and other pesticides.

T

What book did Rachel Carson write in 1962? Silent Spring

Results for SK method

Only SK method on Extra data (460 out of 750)

Methods\ tasks	IE (MUC,BinRel)	QA (TREC2003)	Overall
BoW	62.9%	61.4%	62.3%
TSM	64.9%	62.3%	63.8%
SK	**76.3%**	**65.7%**	**74.5%**

Only SK method on RTE-2 data

Exps\Tasks	IE	IR	QA	SUM	ALL
ExpA1: coverage	**63.3%**	18.3%	**36.3%**	16.3%	**536**
ExpA1: acc. of matches	64%	67.1%	66.2%	73.9%	**66.2%**
ExpA2: coverage	**63.5%**	23.5%	**44%**	17%	**296**
ExpA2: acc. of matches	66.9%	70.2%	58.0%	64.7%	**64.5%**

Comments

☆ **Coverage:**

– For IE and QA pairs, SK+BS reveals a better coverage, more than a half

– For IR and SUM pairs, although it achieves good accuracies, the number of covered cases is low

☆ **Task-based strategy selection:**

– IE and QA: SK+TSM

– IR: SK+BoW

– SUM: BoW

Future Work

☆ RTE core method

– Increase coverage of SK method

  • Integrate IE technology, especially NE recognition
  • Lexical semantics of function words
  • Extend to n-ary hypothesis texts

– Adapt to German language (e.g., rich morphology, noun compounds)

☆ Applications

– Entailment-based QA system on structured data (*QALL-ME*, project funded by European Commission)

– Unsupervised Relation extraction (*IDEX*, project funded by Investionsbank Berlin)