A Multi-Domain Web-Based Algorithm for POS Tagging of Unknown Words

Shulamit Umansky-Pesin, Roi Reichart,
Ari Rappoport

האוניברסיטה העברית בירושלים
The Hebrew University of Jerusalem

Outline

- Introduction
- Algorithm
- Experimental results
- Conclusions

Part-Of-Speech tagging

- The POS tagging problem
  - Determine the POS tag for a particular instance of a word
- Supervised taggers perform well:
  - Toutanova et al., 2003: 97.24% overall accuracy on WSJ corpus
  - But only 89.04% accuracy on unknown words

Domain adaptation

- The training and test corpora are from different domains
- Number of unknown words increases
- The total and unknown words accuracy suffers:
  - Tagging GENIA: 80.12% accuracy on unknown words
  - Tagging BNC: 68.71% accuracy on unknown words

Previous approaches

- Unknown words treatment:
  - Orthographical data (capital letters, digits, hyphens)
  - Prefixes and suffixes
  - Language-specific hand-crafted morphological and syntactic features
  - External data (lexicons etc.)

Previous approaches

- Domain adaptation:
  - Daume III, 2007 – manually labeled corpus from target domain
  - Blitzer et al., 2006 – unlabeled corpus from target domain
- Target domain is not always well-defined (for example, web)

- Preparing a corpus is time-consuming, labeling it is much more so.

Outline

- Introduction
- Algorithm
- Experimental results
- Conclusions

Web search and context

- "You shall know a word by the company it keeps" (John Rupert Firth, 1957)
- Retrieve the "company" from the web
  - Who else "keeps the same company" (replacement)
  - The "company" on one side given the word and "company" on the other side (left-side and right-side contexts)

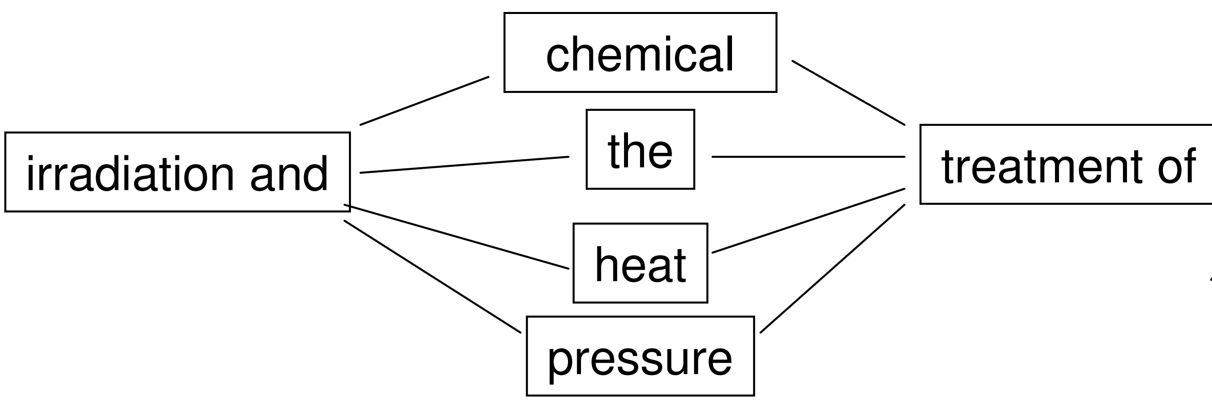
Web search and context

Unknown word

“UV irradiation and **H2O2** treatment of T lymphocytes …”

“irradiation and * treatment of”

Replacement in context



Wouldn’t work alone!

Web search and context

Unknown word

"UV irradiation and **H2O2** treatment of T lymphocytes …"

Left-side context

"* * H2O2 treatment of"

by an
indicated that
enhanced by
familiar with
observed after

H2O2 treatment of

Right-side context

"irradiation and H2O2 * *"

irradiation and H2O2

on comparison
on Fe
treatment by
cause an
does not

POS tagger

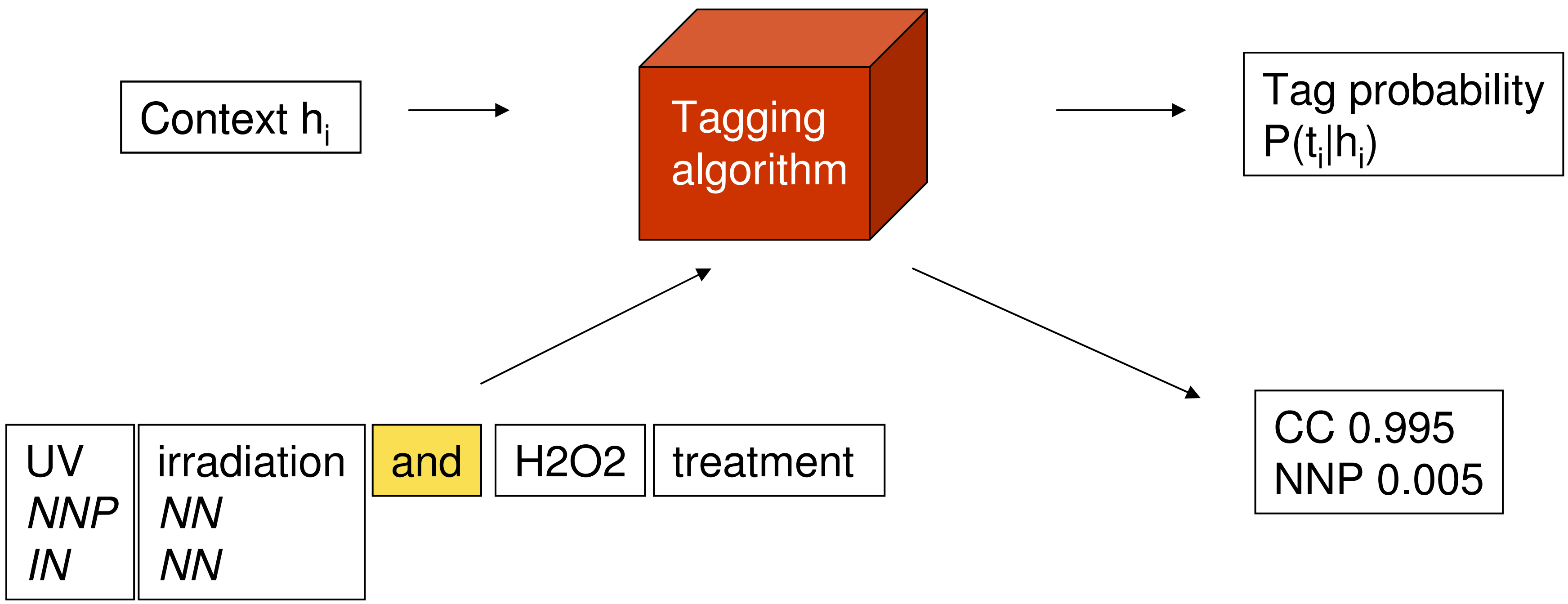
- Maximum Entropy tagger - reimplementation of MxPOST (Ratnaparkhi, 1996)
- Training phase left unchanged
- Original (Ratnaparkhi, 1996) features used
- POS tag is determined by 2-words context

MaxEnt features

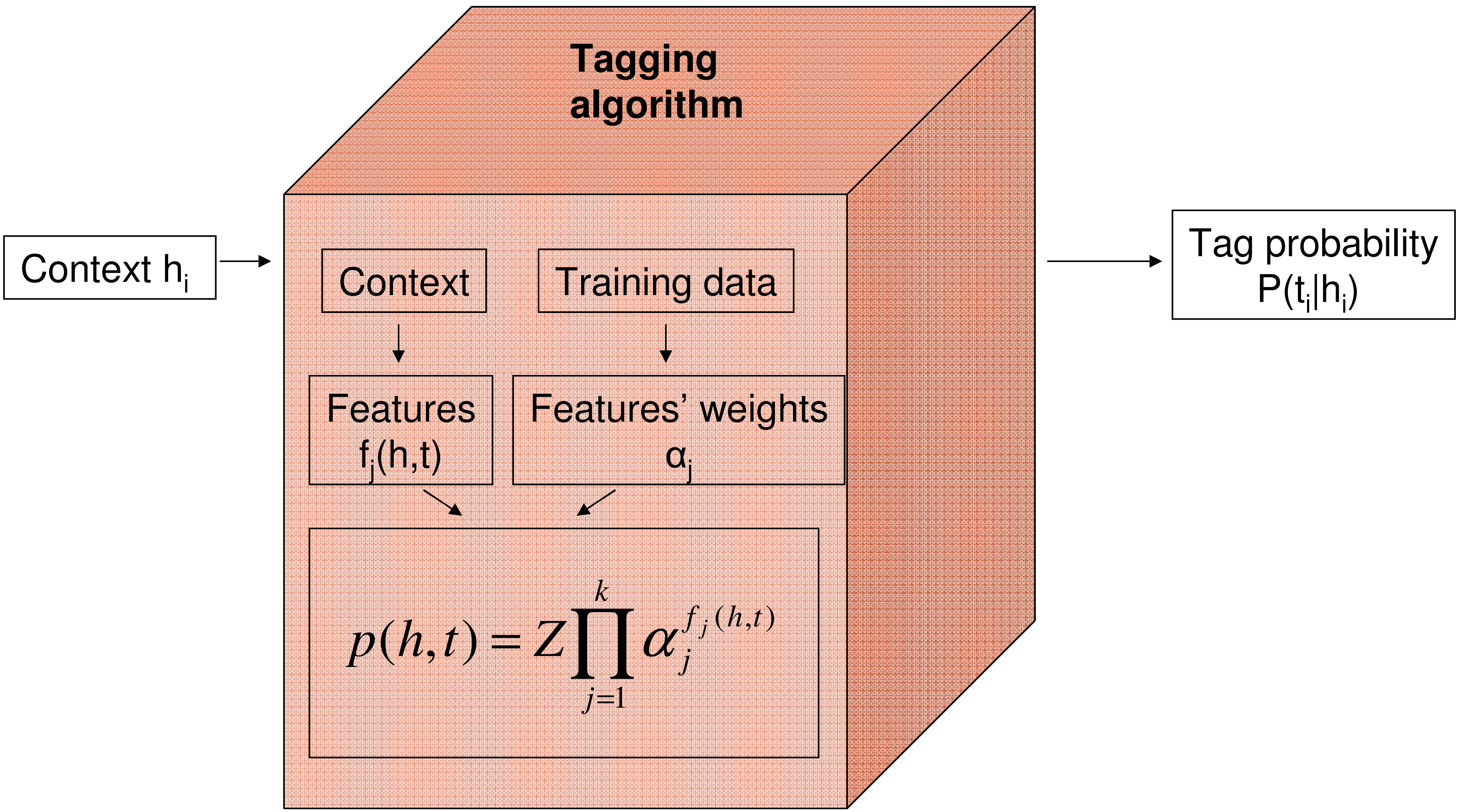
Condition	Features	
w_i is not rare	$w_i = X$	$\& \; t_i = T$
w_i is rare	X is prefix of w_i, $\lvert X \rvert \leq 4$	$\& \; t_i = T$
	X is suffix of w_i, $\lvert X \rvert \leq 4$	$\& \; t_i = T$
	w_i contains number	$\& \; t_i = T$
	w_i contains uppercase character	$\& \; t_i = T$
	w_i contains hyphen	$\& \; t_i = T$
$\forall \; w_i$	$t_{i-1} = X$	$\& \; t_i = T$
	$t_{i-2}t_{i-1} = XY$	$\& \; t_i = T$
	$w_{i-1} = X$	$\& \; t_i = T$
	$w_{i-2} = X$	$\& \; t_i = T$
	$w_{i+1} = X$	$\& \; t_i = T$
	$w_{i+2} = X$	$\& \; t_i = T$

Table 1: Features on the current history h_i

MaxEnt tagger - reminder



MaxEnt tagger - reminder



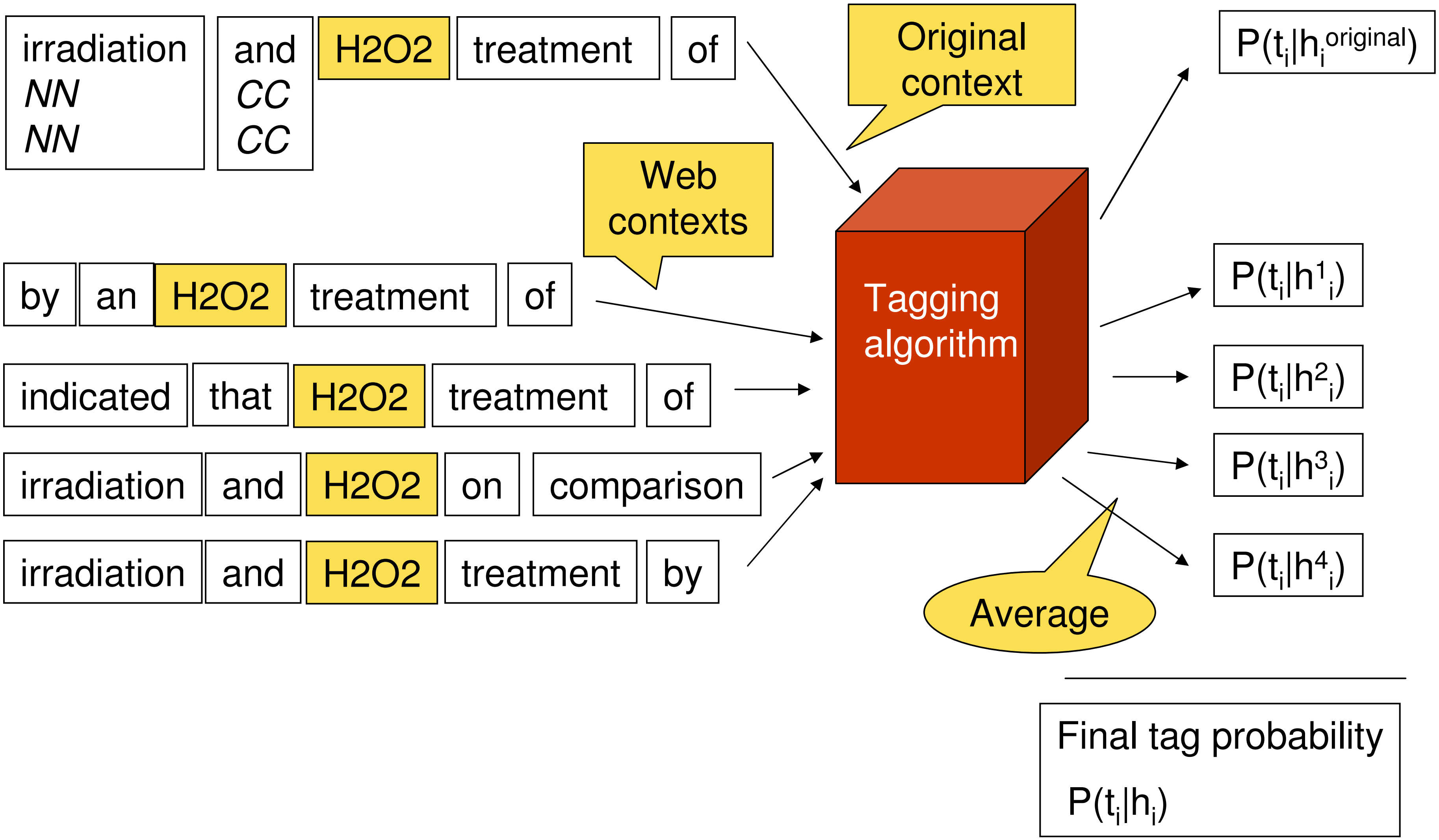
MaxEnt - reminder

UV *NNP* *IN*	irradiation *NN* *NN*	and	H2O2	treatment	of	…

- At each step maintain a list N of tag sequences:
  - *UV_NNP irradiation_NN*
  - *UV_NNP irradiation_NNP*
- For each candidate sequence of tags
  - Extract features for the new word (“*and*”)
  - For each possible* tag
    - Calculate tag conditional probability $P(t_i|h_i)$ using the features parameters learned in training
    - Calculate sequence conditional probability $P(t_1.. t_i|h_1..h_i)$
- Select N top-scoring sequences
- Repeat

***possible tags:**
•All tags for *unknown* words
•Only tags seen in training for *known* words

Unknown words & web search



Unknown words & web search

UV	irradiation	and	H2O2	treatment	of	…
NNP	*NN*	*CC*				
NNP	*JJ*	*CC*				

- Additional steps:
  - Collect left- and right-side contexts and replacements from the web and create new words sequences
  - For each new words sequence h'_i
    - For each tag
      - Calculate tag conditional probability $P(t_i|h'_i)$ using the features from the new context
  - Calculate final tag probability as the average between all $P(t_i|h'_i)$ and the original $P(t_i|h_i)$

Outline

- Introduction
- Algorithm
- Experimental results
- Conclusions

Experimental setup

- Unknown words threshold: 5
- Baseline: MxPOST tagger

Experimental setup - English

Name	Training	Testing
WSJ	WSJ 2-21	WSJ 23
GENIA (domain adaptation)	WSJ 2-21	2000 sentences sample from GENIA
BNC (domain adaptation)	WSJ 2-21	2000 sentences sample from BNC

Results - English

Unknown words accuracy

	WSJ	GENIA	BNC
Baseline	*88.79%*	*80.12%*	*68.71%*
Web-assisted	**89.86%**	**83.00%**	**72.12%**
Improvement	1.07%	2.88%	3.41%
Error reduction	**9.54%**	**14.48%**	**10.89%**

Experimental setup - German

Name	Training	Testing
Negra	15689 NEGRA sentences	2096 NEGRA sentences
Tiger (domain adaptation)	15689 NEGRA sentences	2000 TIGER sentences
Negra (domain adaptation)	15689 TIGER sentences	2096 NEGRA sentences

Results - German

Unknown words accuracy

	Negra	Tiger domain adaptation	Negra domain adaptation
Baseline	*91.06%*	*87.88%*	*87.86%*
Web-assisted	**91.95%**	**89.01%**	**89.84%**
Improvement	0.89%	1.13%	1.98%
Error reduction	**9.95%**	**9.32%**	**16.3%**

Experimental setup - Chinese

Name	Training	Testing
CTB	14903 CTB sentences	1945 CTB sentences

Results - Chinese

Unknown words accuracy

	CTB
Baseline	*78.03%*
Web-assisted	**80.75%**
Improvement	2.72%
Error reduction	**12.28%**

Outline

- Introduction
- Algorithm
- Experimental results
- Conclusions

Conclusions

- No preprocessing steps!
- Train once, tag anything – no knowledge about domain is required
- Language-independent
- Can be adapted to suit other taggers

What about Hebrew?

What about Hebrew?

- Some additional segmentation of web matches is required
- Other than that… should work!

Thank you