Multiword Expression Identification with Tree Substitution Grammars

Spence Green, Marie-Catherine de Marneffe,
John Bauer, and Christopher D. Manning

Stanford University

EMNLP 2011

Main Idea

Use **syntactic context** to find multiword expressions

Main Idea

Use **syntactic context** to find multiword expressions

Syntactic context $\rightarrow$ constituency parses

Main Idea

Use **syntactic context** to find multiword expressions

Syntactic context $\rightarrow$ constituency parses

Multiword expressions $\rightarrow$ idiomatic constructions

Which languages?

Results and analysis for **French**

Which languages?

Results and analysis for **French**

- Lexicographic tradition of compiling MWE lists
- Annotated data!

Which languages?

Results and analysis for **French**

- Lexicographic tradition of compiling MWE lists
- Annotated data!

English examples in the talk

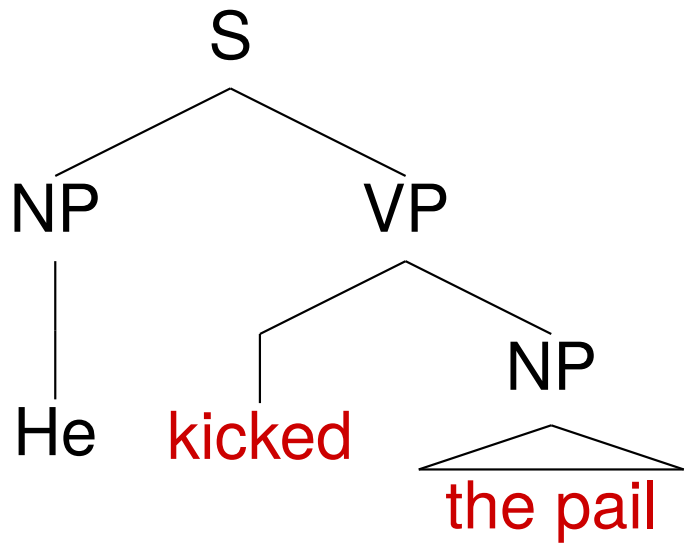
Motivating Example: Humans get this

1. He kicked the pail.

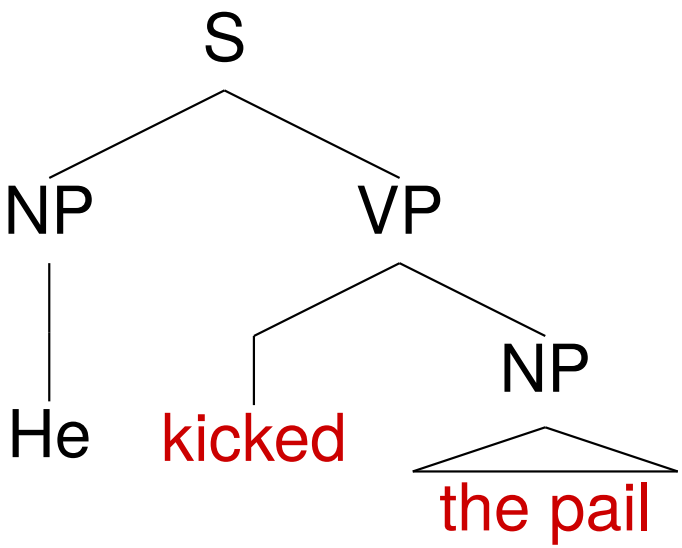
2. He kicked the bucket.
   - “He died.”

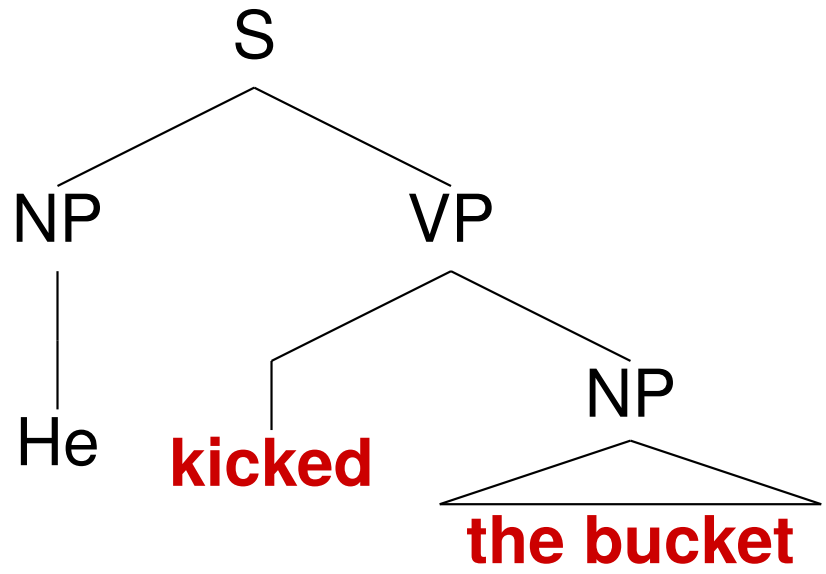
(Katz and Postal 1963)

Stanford parser can't tell the difference



Stanford parser can't tell the difference





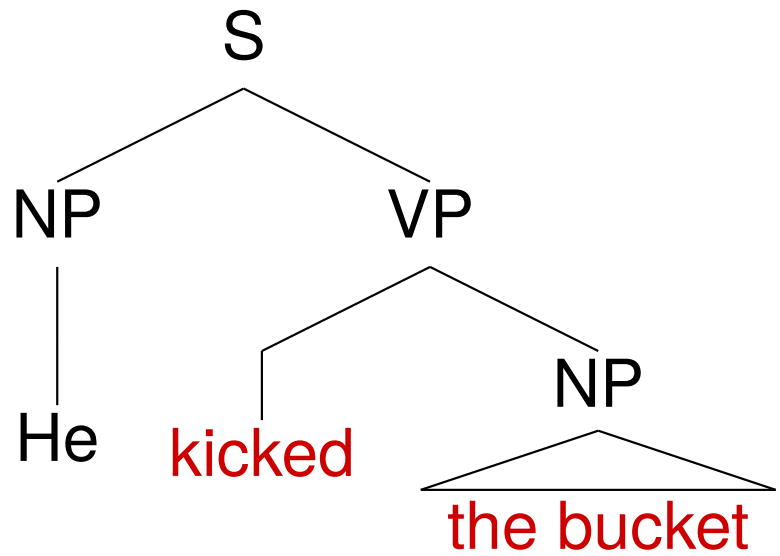
What does the lexicon contain?

Single-word entries?

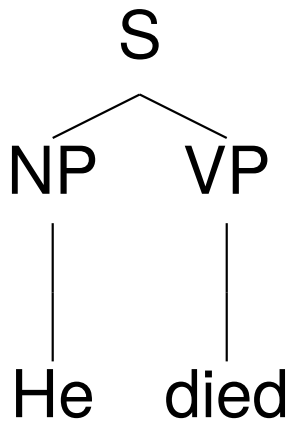
- *kick* : <agent, theme>
- *die* : <theme>

Multi-word entries?

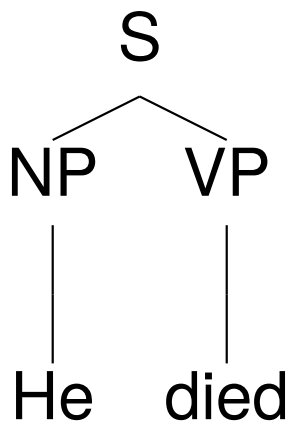
- *kick the bucket* : <theme>

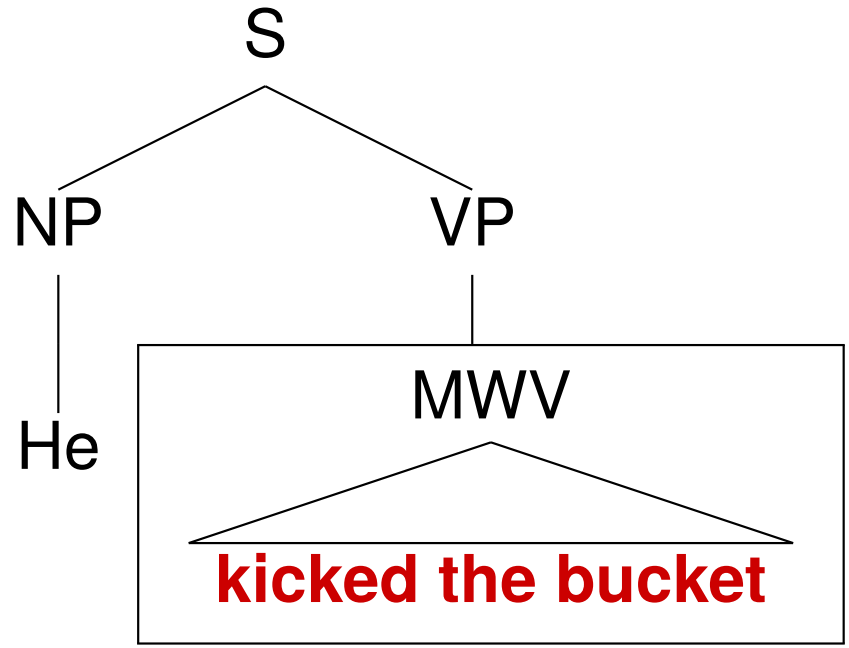


Lexicon-Grammar: *He kicked the bucket*



Lexicon-Grammar: *He kicked the bucket*





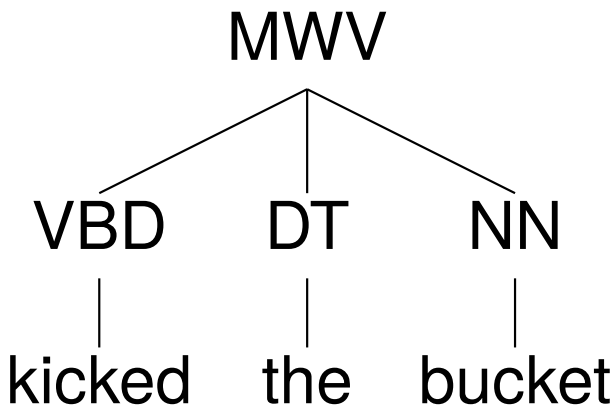
(Gross 1986)

MWEs in Lexicon-Grammar

Classified by **global** POS

Described by **internal** POS sequence

Flat structures!

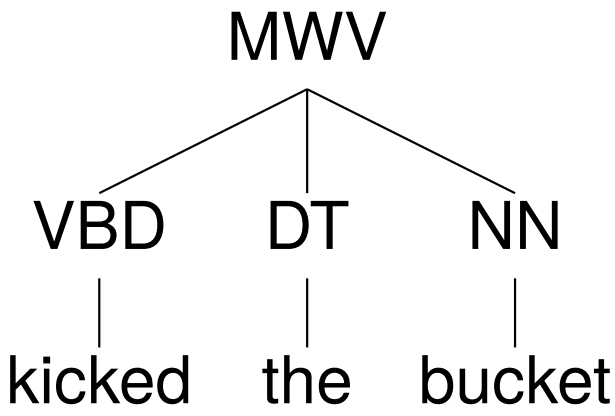


MWEs in Lexicon-Grammar

Classified by **global** POS

Described by **internal** POS sequence

Flat structures!



Of theoretical interest but...

Why do we care (in NLP)?

MWE knowledge improves:

Dependency parsing	(Nivre and Nilsson 2004)
Constituency parsing	(Arun and Keller 2005)
Sentence generation	(Hogan et al. 2007)
Machine translation	(Carpuat and Diab 2010)
Shallow parsing	(Korkontzelos and Manandhar 2010)

Why do we care (in NLP)?

MWE knowledge improves:

Dependency parsing	(Nivre and Nilsson 2004)
Constituency parsing	(Arun and Keller 2005)
Sentence generation	(Hogan et al. 2007)
Machine translation	(Carpuat and Diab 2010)
Shallow parsing	(Korkontzelos and Manandhar 2010)

Most experiments assume high accuracy identification!

French and the French Treebank

MWEs common in French

- $\sim$5,000 multiword adverbs

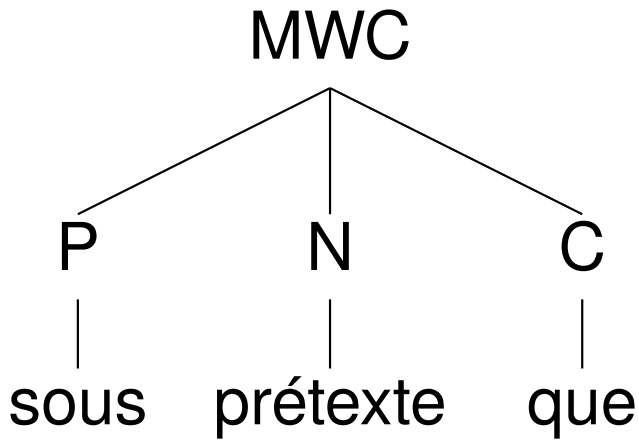
French and the French Treebank

MWEs common in French

- ∼5,000 multiword adverbs

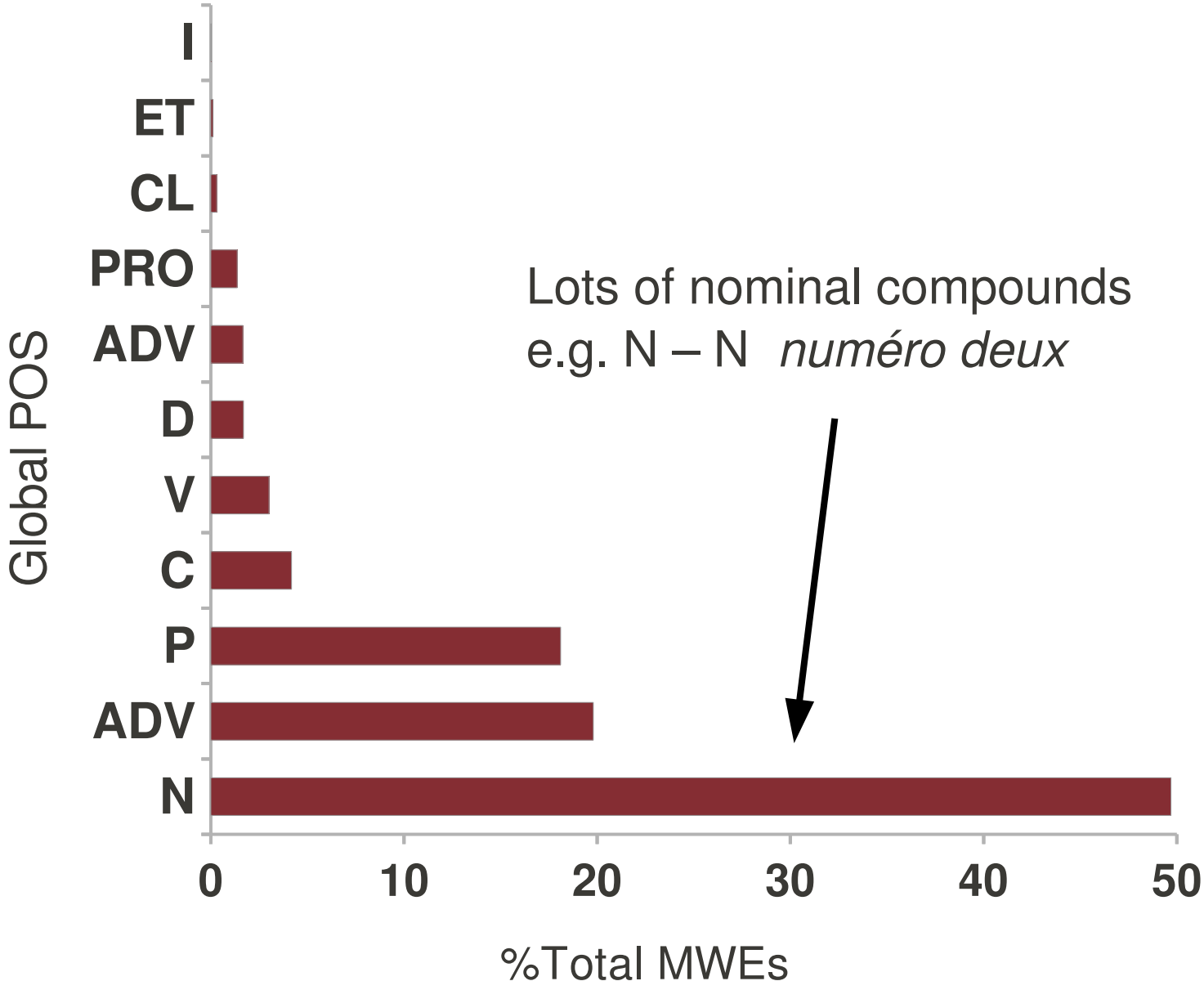
Paris 7 French Treebank

- ∼16,000 trees
- 13% of tokens are MWE



on the grounds that

French Treebank: MWE types



MWE Identification Evaluation

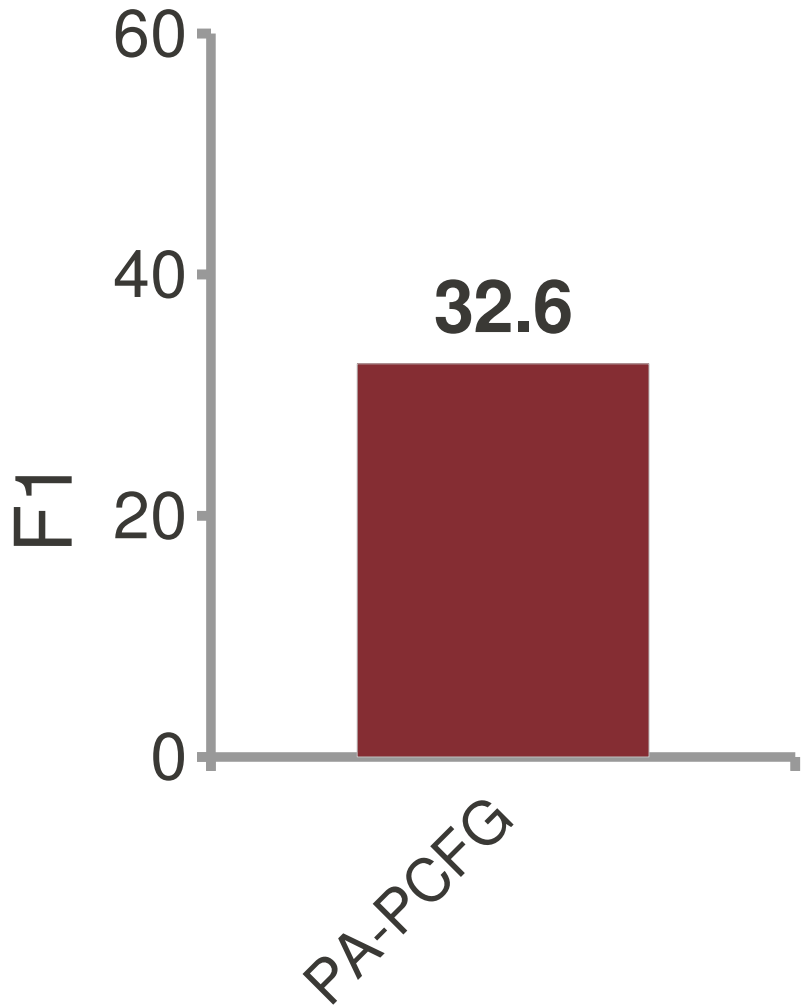
Identification is a by-product of parsing

MWE Identification Evaluation

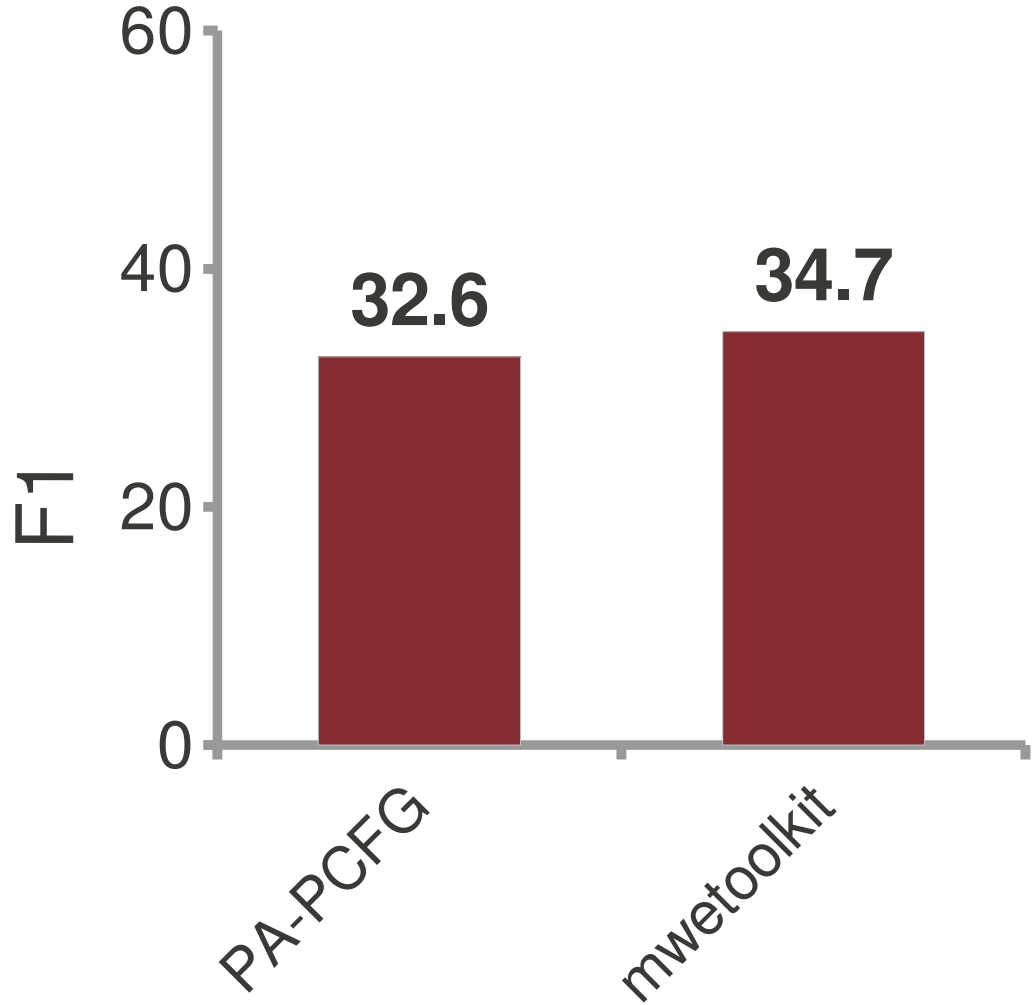
Identification is a by-product of parsing

- Corpus: Paris 7 French Treebank (FTB)
- Split: same as (Crabbé and Candito 2008)
- Metrics: Precision and Recall
- Lengths $\leq$ 40 words

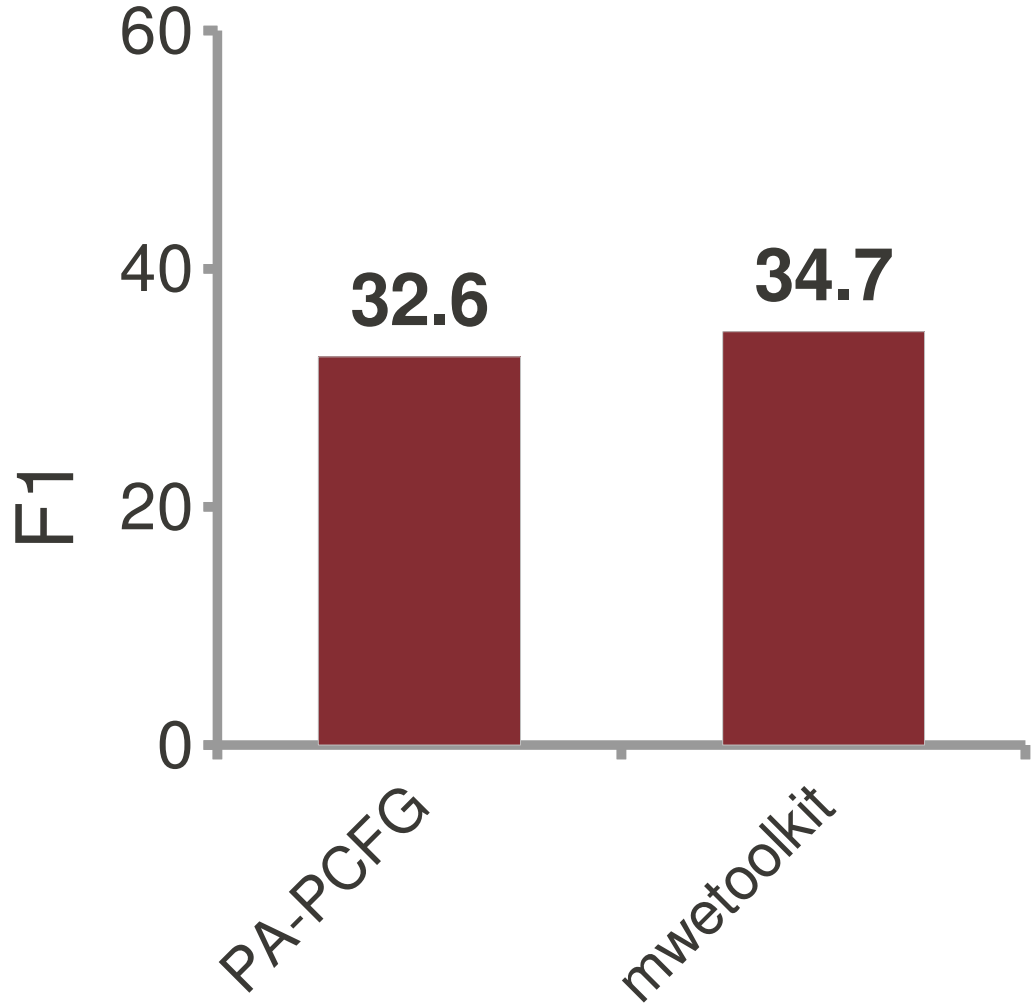
MWE Identification: Parent-Annotated PCFG



MWE Identification: *n*-gram methods



MWE Identification: *n*-gram methods



Standard approach in 2008 MWE Shared Task, MWE Workshops, etc.

n-gram methods: `mwetoolkit`

Based on surface statistics

n-gram methods: `mwetoolkit`

Based on surface statistics

Step 1: Lemmatize and POS tag corpus

n-gram methods: `mwetoolkit`

Based on surface statistics

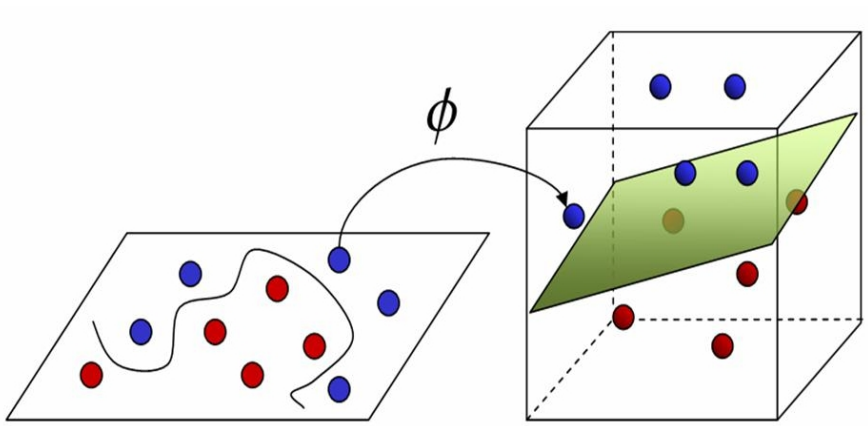
Step 1: Lemmatize and POS tag corpus

Step 2: Compute *n*-gram statistics:

- Maximum likelihood estimator
- Dice's coefficient
- Pointwise mutual information
- Student's *t*-score

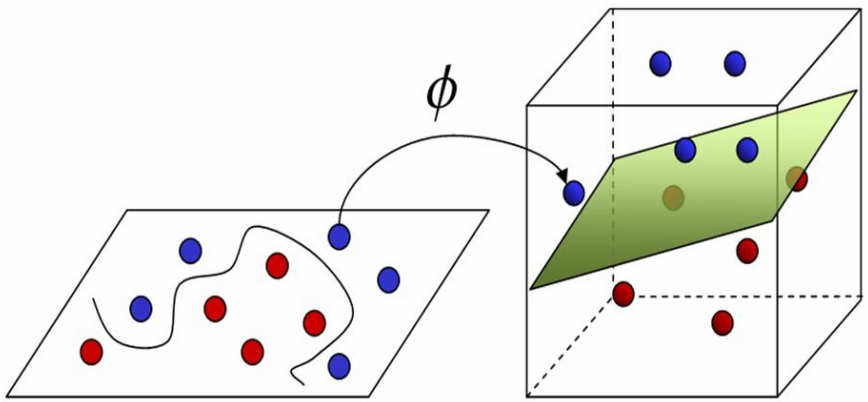
(Ramisch, Villavicencio, and Boitet 2010)

n-gram methods: `mwetoolkit`



Step 3: Create *n*-gram feature vectors

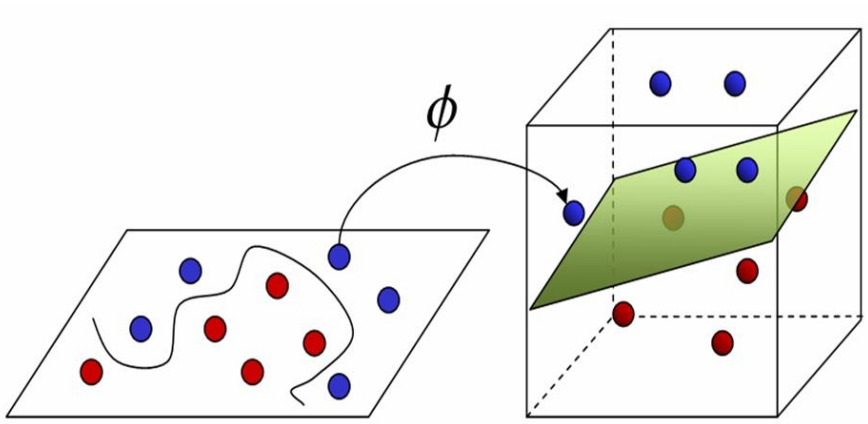
n-gram methods: `mwetoolkit`



Step 3: Create *n*-gram feature vectors

Step 4: Train a binary classifier

n-gram methods: `mwetoolkit`



Step 3: Create *n*-gram feature vectors

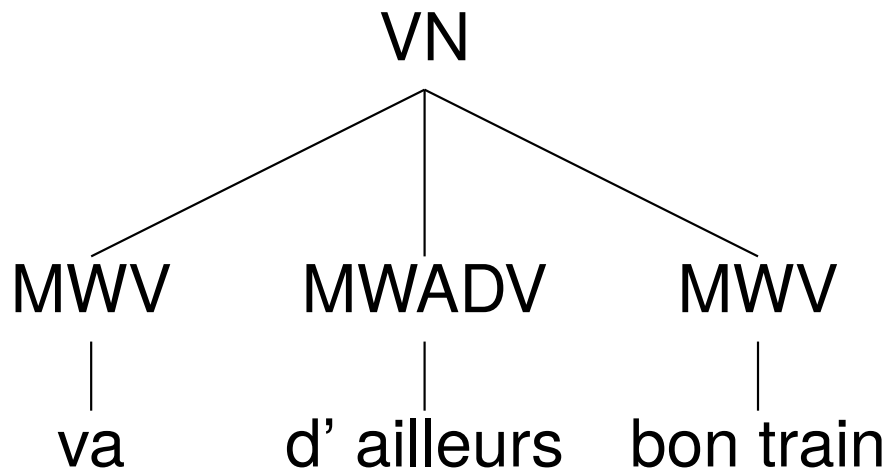
Step 4: Train a binary classifier

Exploits statistical idiomaticity of MWEs

Is statistical idiomaticity sufficient?

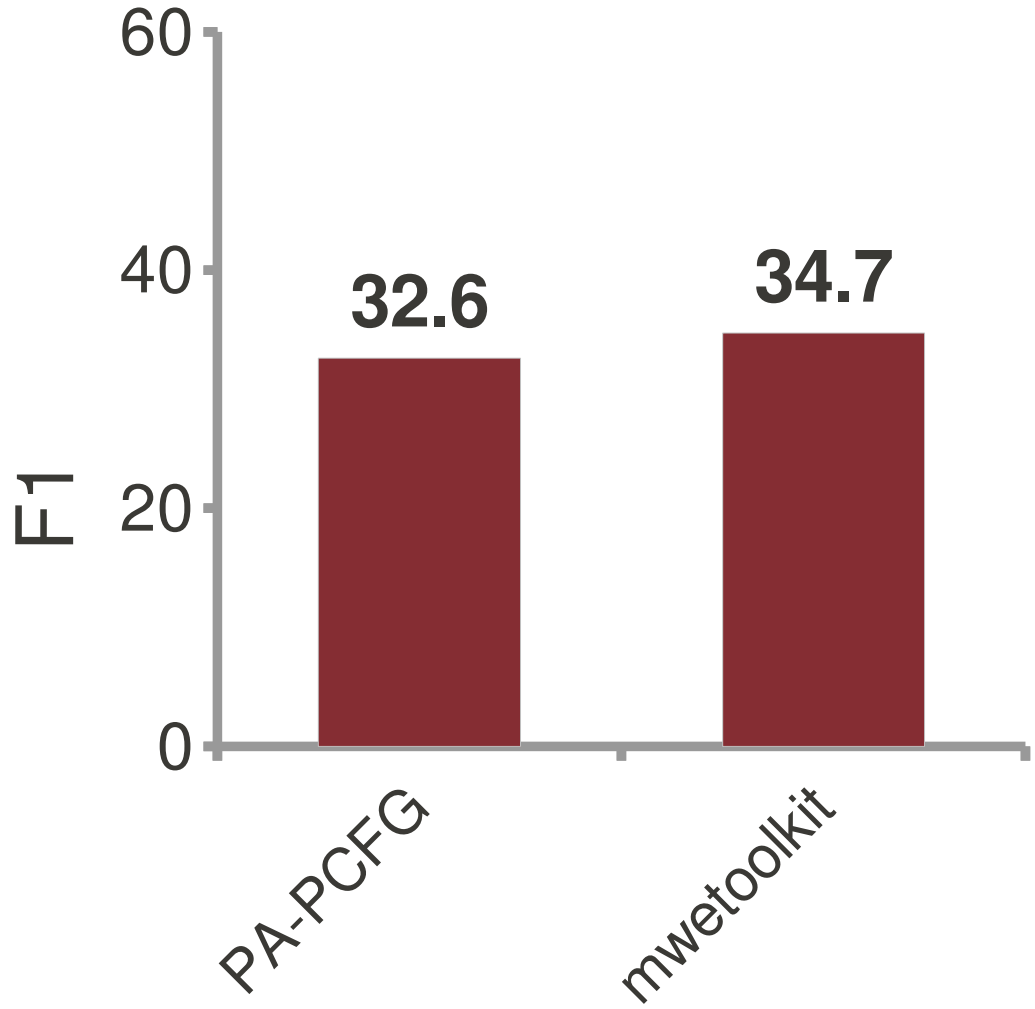
French multiword verbs

Tree maintains relationship between MWV parts

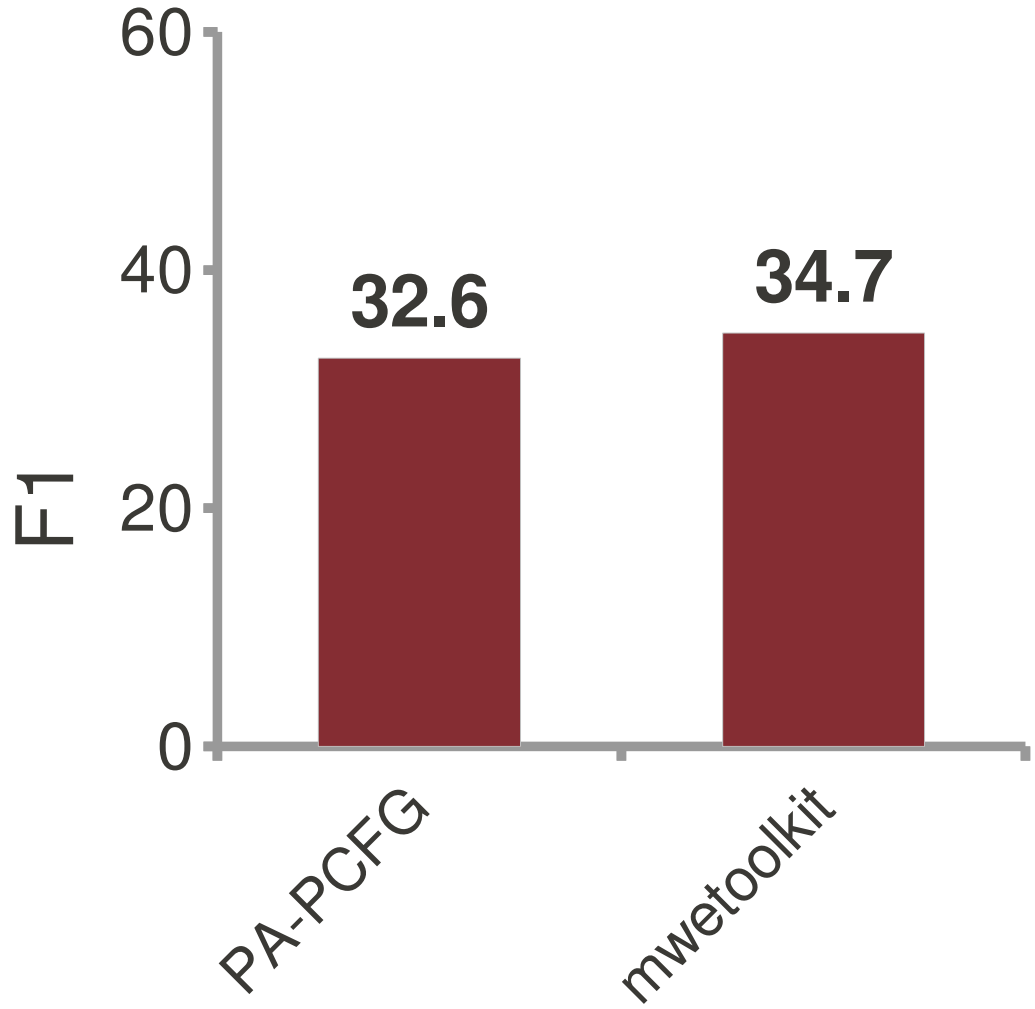


is also well underway

Recap: French MWE Identification Baselines



Recap: French MWE Identification Baselines



Let's build a better grammar

Better PCFGs: Manual grammar splits

Symbol refinement à la (Klein and Manning 2003)

Better PCFGs: Manual grammar splits

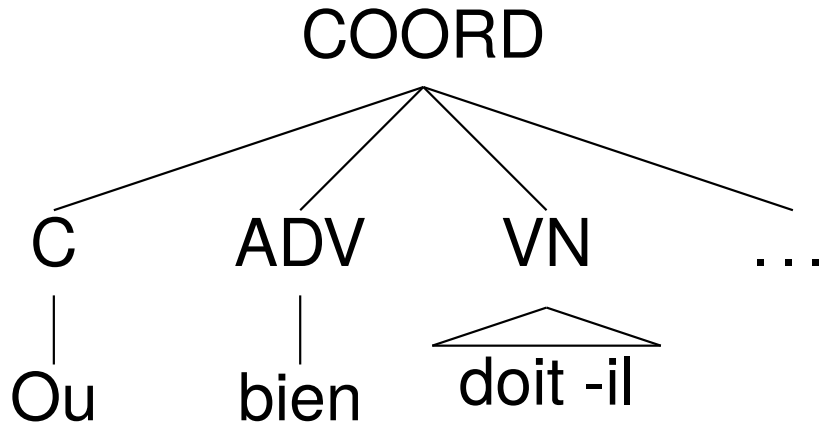
Symbol refinement à la (Klein and Manning 2003)

- Has a verbal nucleus (VN)

Better PCFGs: Manual grammar splits

Symbol refinement à la (Klein and Manning 2003)

- Has a verbal nucleus (VN)

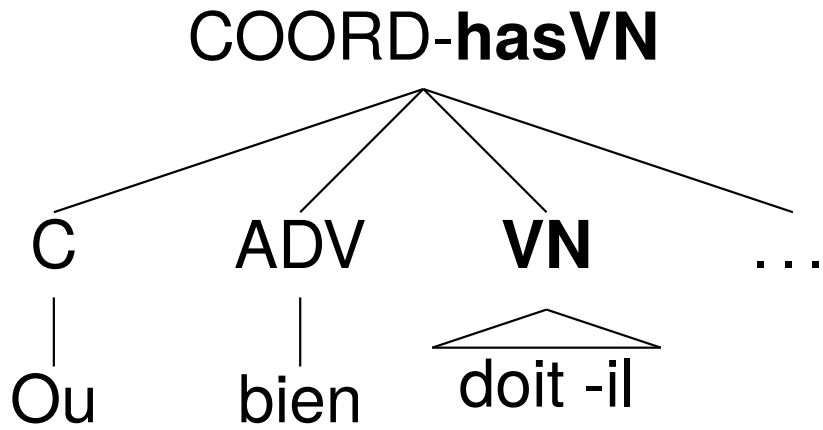


Otherwise he must

Better PCFGs: Manual grammar splits

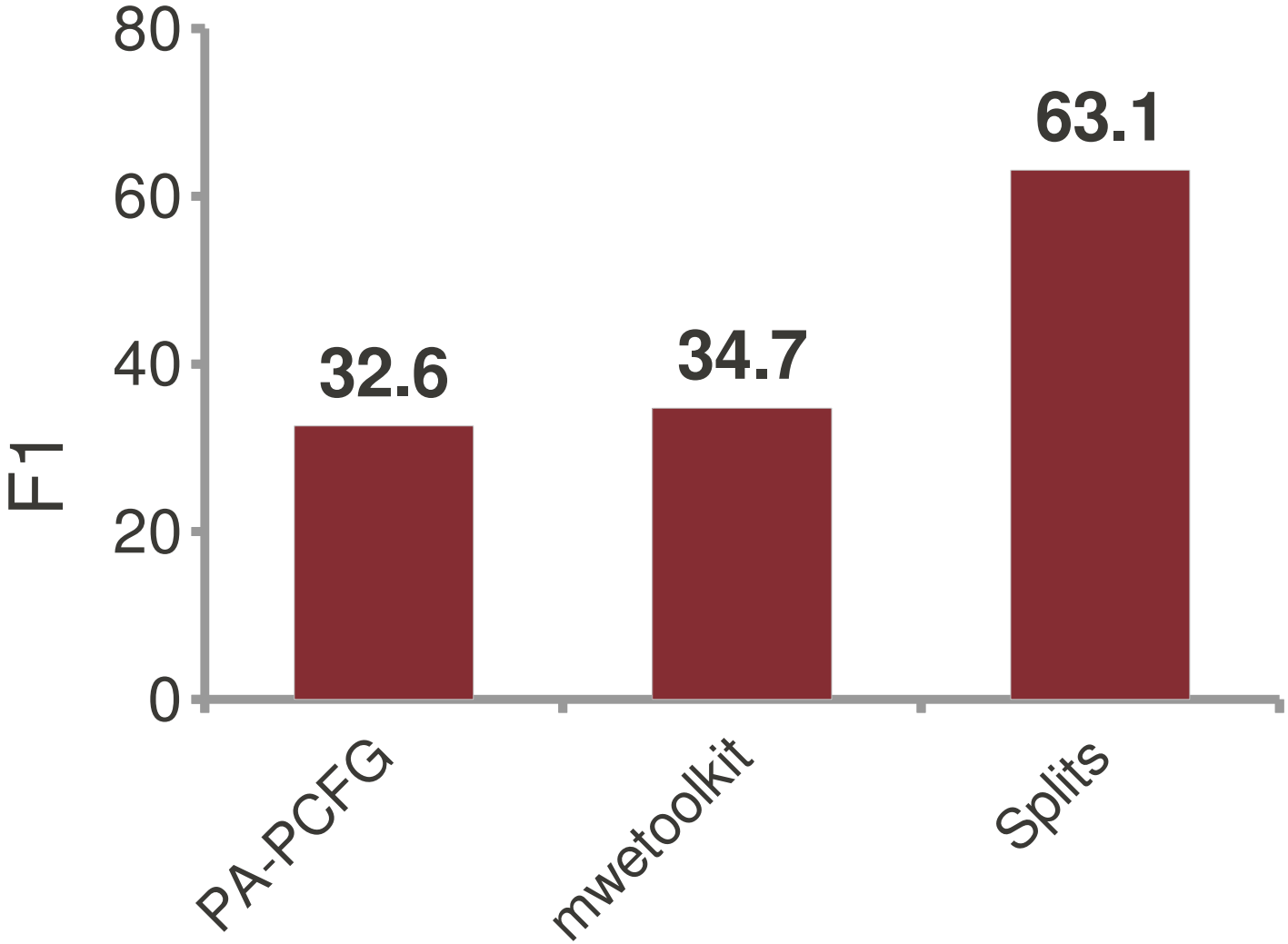
Symbol refinement à la (Klein and Manning 2003)

- Has a verbal nucleus (VN)

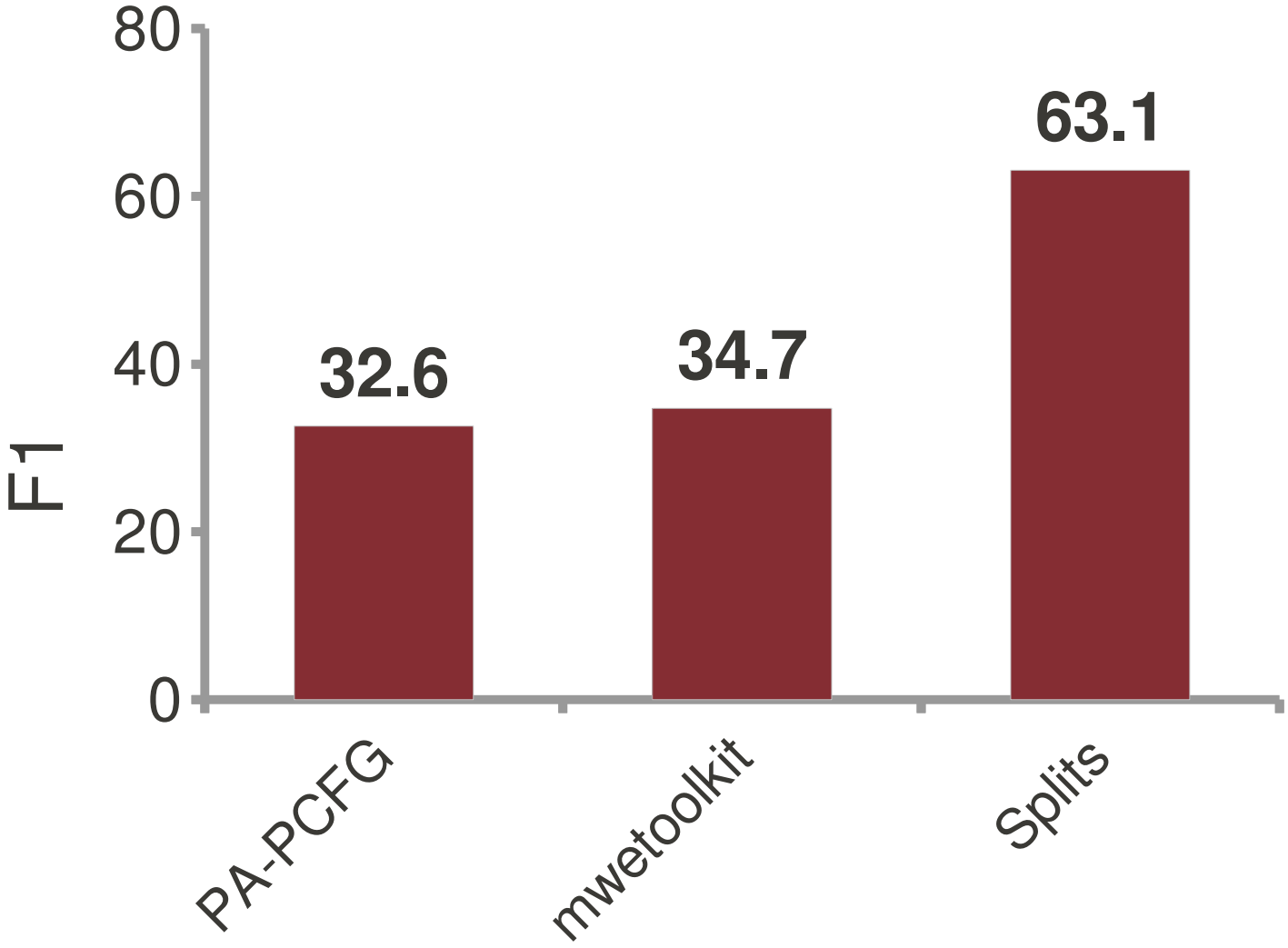


Otherwise he must

French MWE Identification: Manual Splits



French MWE Identification: Manual Splits



MWE features: high frequency POS sequences

Capture more syntactic context?

PCFGs work well!

Capture more syntactic context?

PCFGs work well!

Larger “rules”: **Tree Substitution Grammars** (TSG)

Capture more syntactic context?

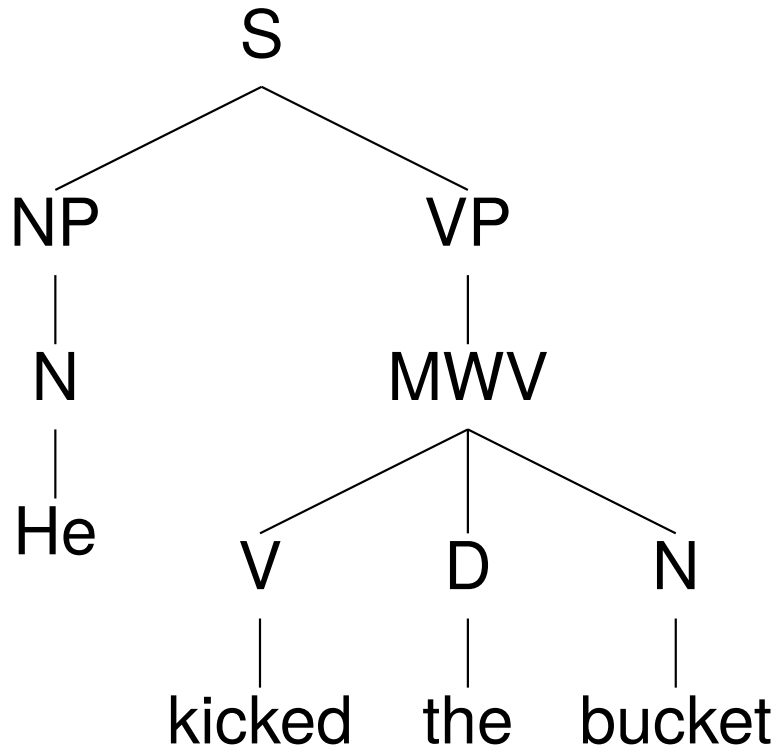
PCFGs work well!

Larger “rules”: **Tree Substitution Grammars** (TSG)

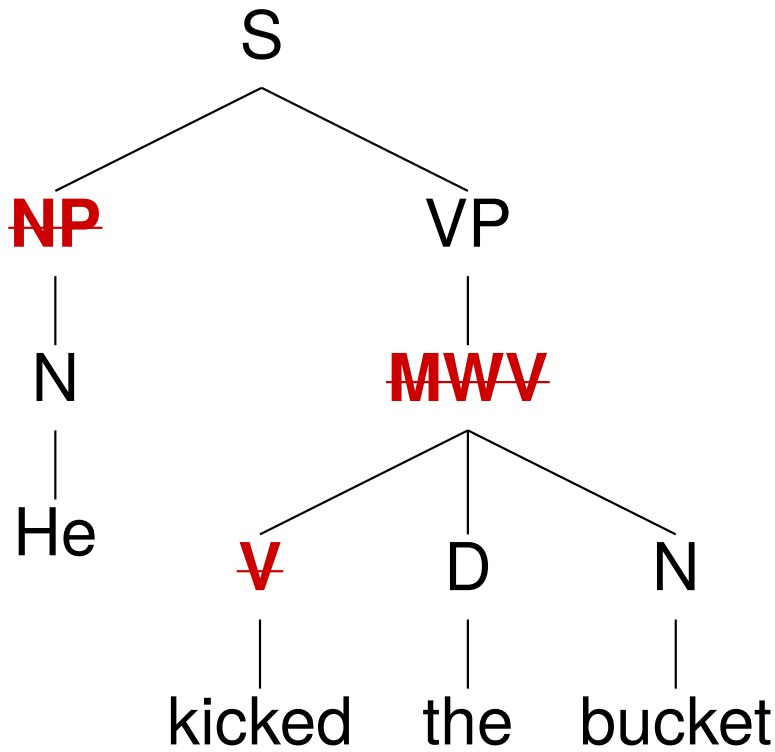
Relationship with Data-Oriented Parsing (DOP):

- Same grammar formalism (TSG)
- We include unlexicalized fragments
- Different parameter estimation

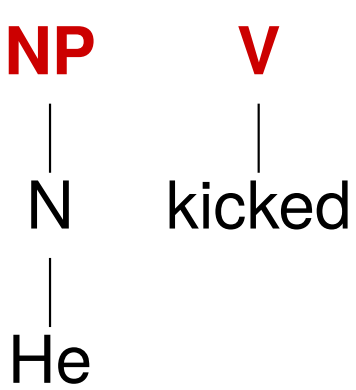
Which tree fragments do we select?

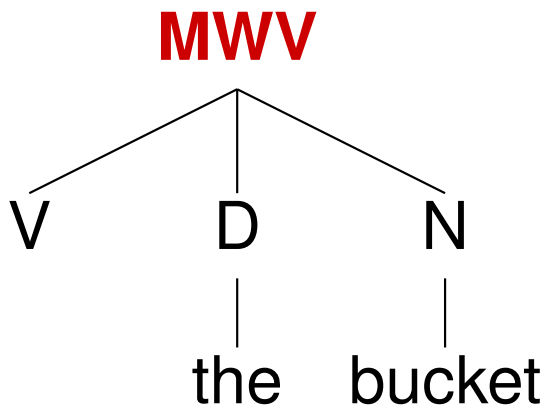


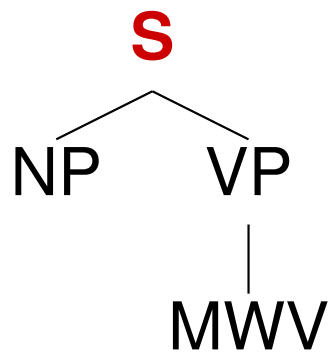
Which tree fragments do we select?



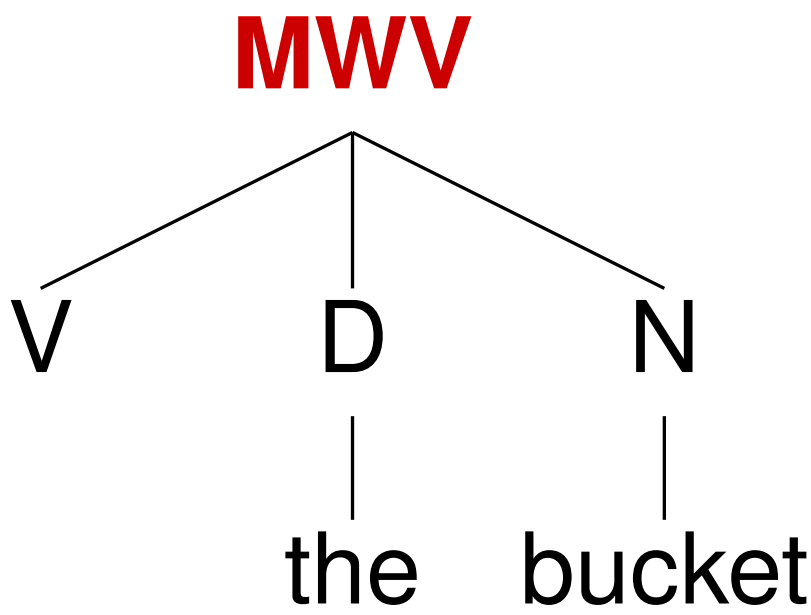
Which tree fragments do we select?



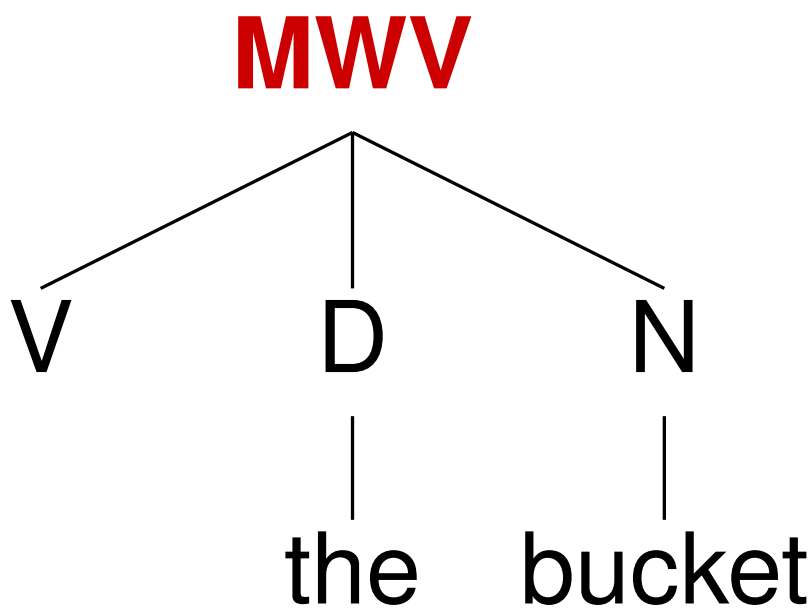




TSG Grammar Extraction as Tree Selection



TSG Grammar Extraction as Tree Selection



- Describes MWE context
- Allows for inflection: *kick, kicked, kicking*

Dirichlet process TSG (DP-TSG)

Tree selection as non-parametric clustering[1]

[1] Cohn, Goldwater, and Blunsom 2009; Post and Gildea 2009; O'Donnell, Tenenbaum, and Goodman 2009.

Dirichlet process TSG (DP-TSG)

Tree selection as non-parametric clustering[1]

Labeled Chinese Restaurant process

- *Dirichlet process* (DP) prior for each non-terminal type c

[1] Cohn, Goldwater, and Blunsom 2009; Post and Gildea 2009; O'Donnell, Tenenbaum, and Goodman 2009.

Dirichlet process TSG (DP-TSG)

Tree selection as non-parametric clustering[1]

Labeled Chinese Restaurant process

- *Dirichlet process* (DP) prior for each non-terminal type c

Supervised case: segment the treebank

[1] Cohn, Goldwater, and Blunsom 2009; Post and Gildea 2009; O'Donnell, Tenenbaum, and Goodman 2009.

DP-TSG: Learning and Inference

DP base distribution from manually-split CFG

DP-TSG: Learning and Inference

DP base distribution from manually-split CFG

Type-based Gibbs sampler (Liang, Jordan, and Klein 2010)

- *Fast* convergence: 400 iterations

DP-TSG: Learning and Inference

DP base distribution from manually-split CFG

Type-based Gibbs sampler (Liang, Jordan, and Klein 2010)

- *Fast* convergence: 400 iterations

Derivations of a TSG are a CFG forest

DP-TSG: Learning and Inference

DP base distribution from manually-split CFG

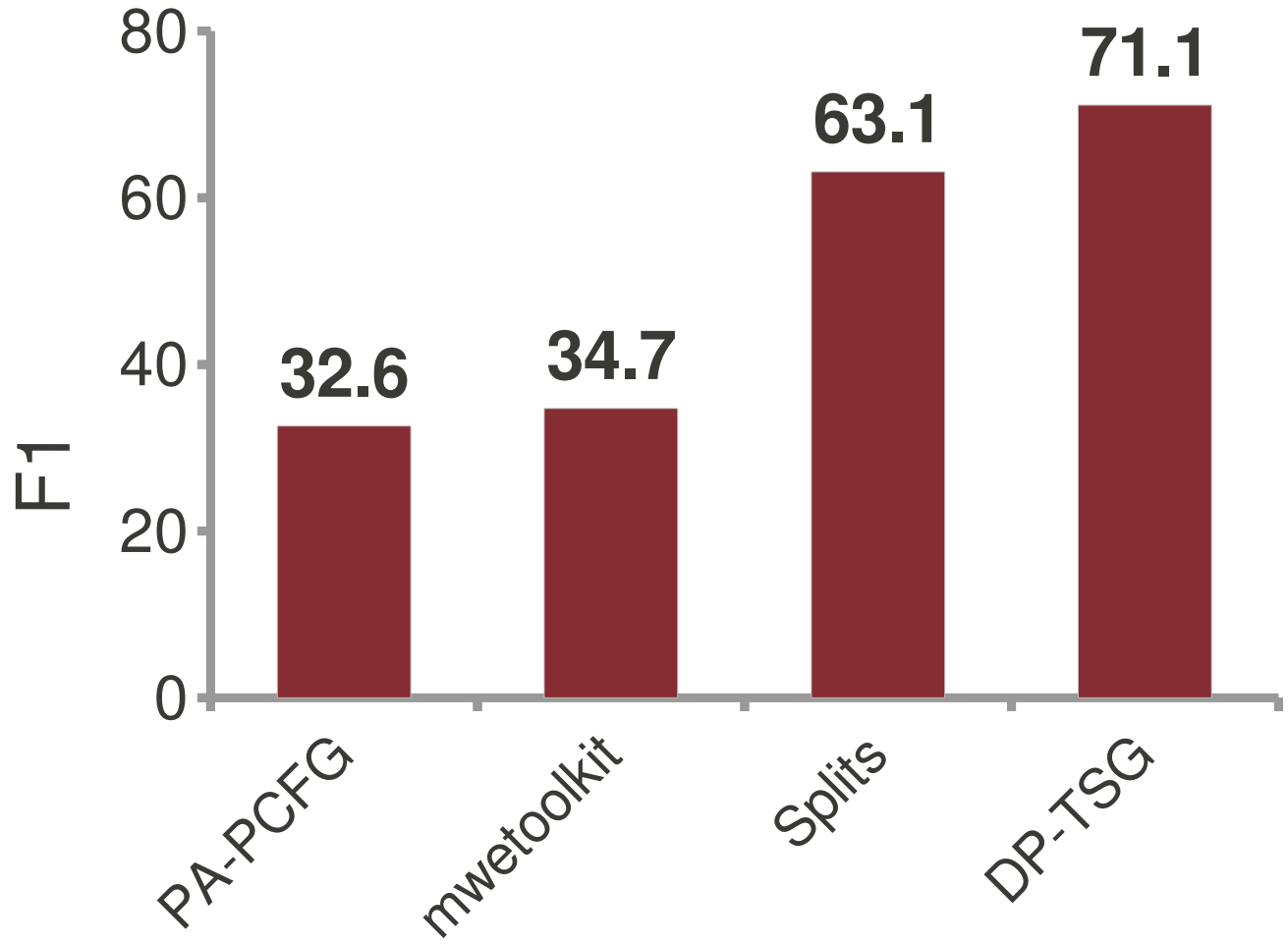
Type-based Gibbs sampler (Liang, Jordan, and Klein 2010)

- *Fast* convergence: 400 iterations

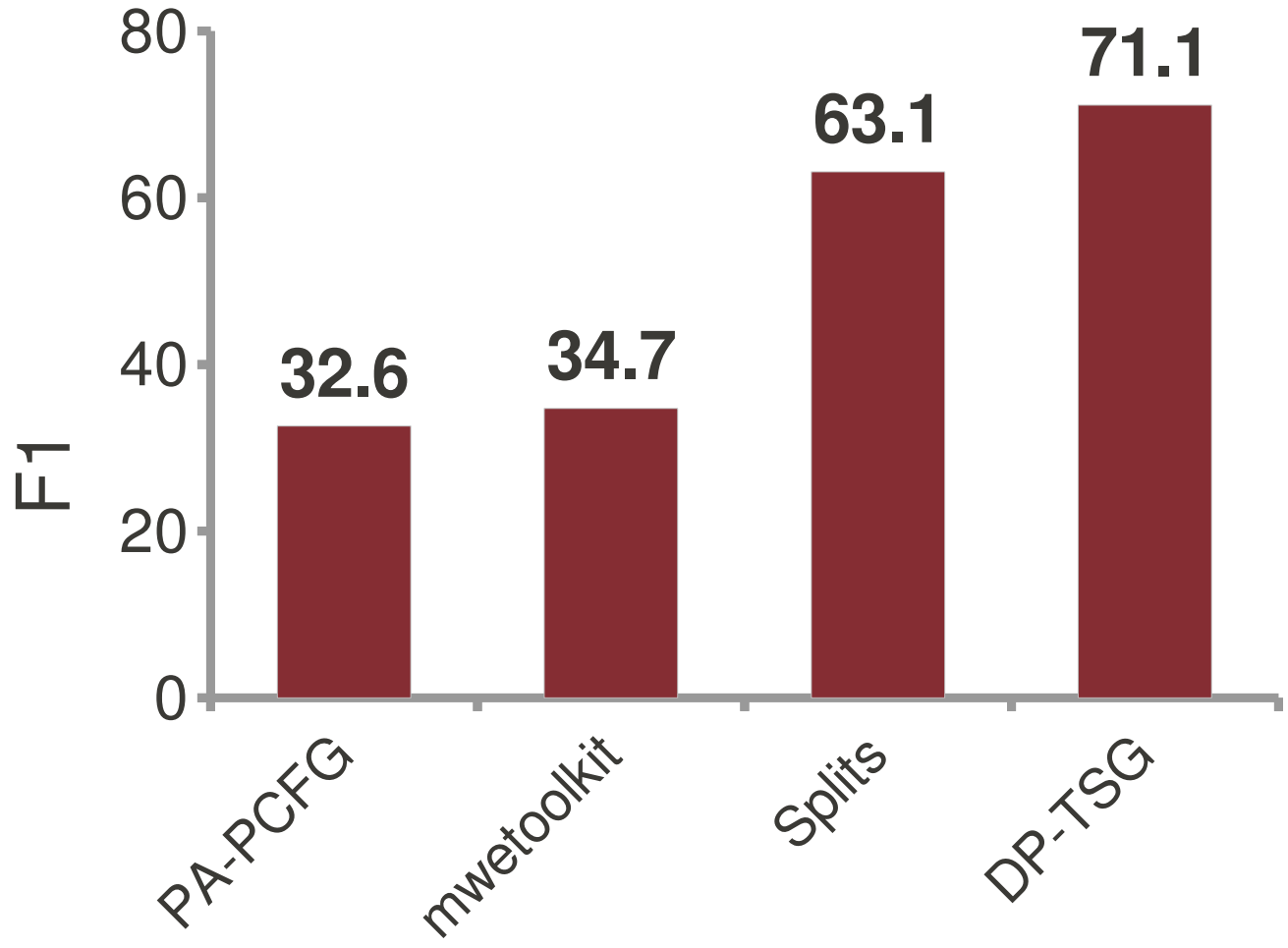
Derivations of a TSG are a CFG forest

- SCFG decoder: `cdec` (Dyer et al. 2010)

French MWE Identification: DP-TSG



French MWE Identification: DP-TSG



DP-TSG result is a lower bound

Human-interpretable DP-TSG rules

MWN $\rightarrow$ coup de N

coup de pied	‘kick’
coup de coeur	‘favorite’
coup de foudre	‘love at first sight’
coup de main	‘help’
coup de grâce	‘death blow’

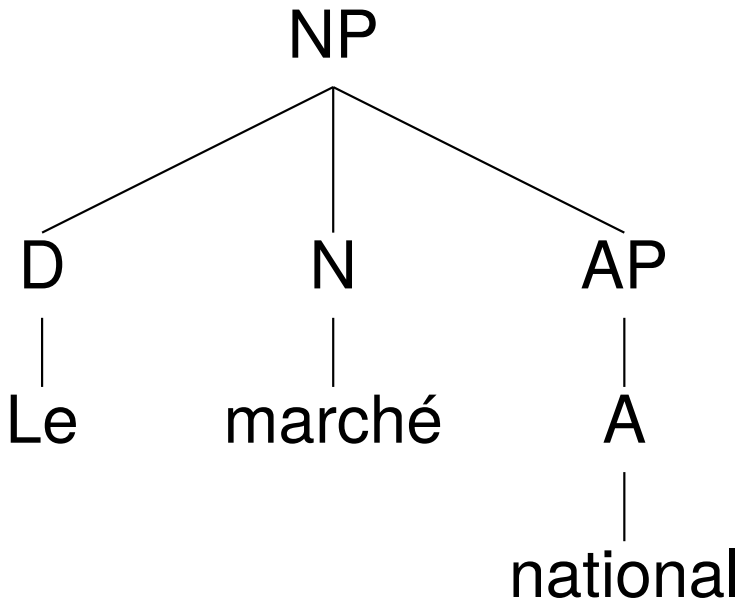
Human-interpretable DP-TSG rules

MWN $\rightarrow$ coup de N

coup de pied	‘kick’
coup de coeur	‘favorite’
coup de foudre	‘love at first sight’
coup de main	‘help’
coup de grâce	‘death blow’

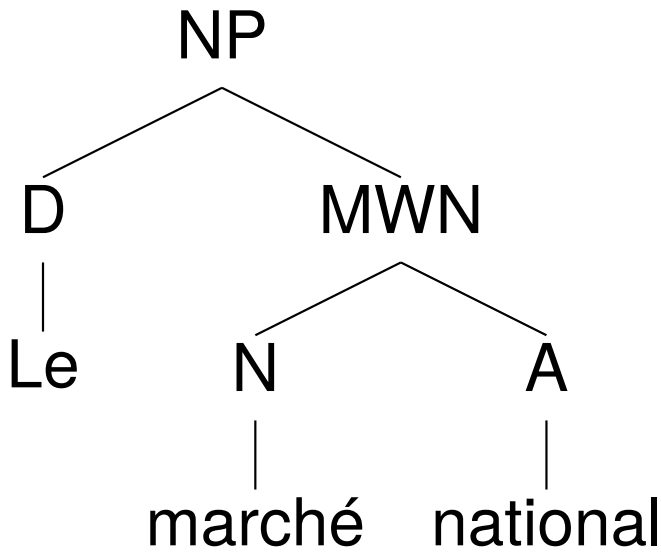
n-gram methods: separate feature vectors

DP-TSG errors: Overgeneration



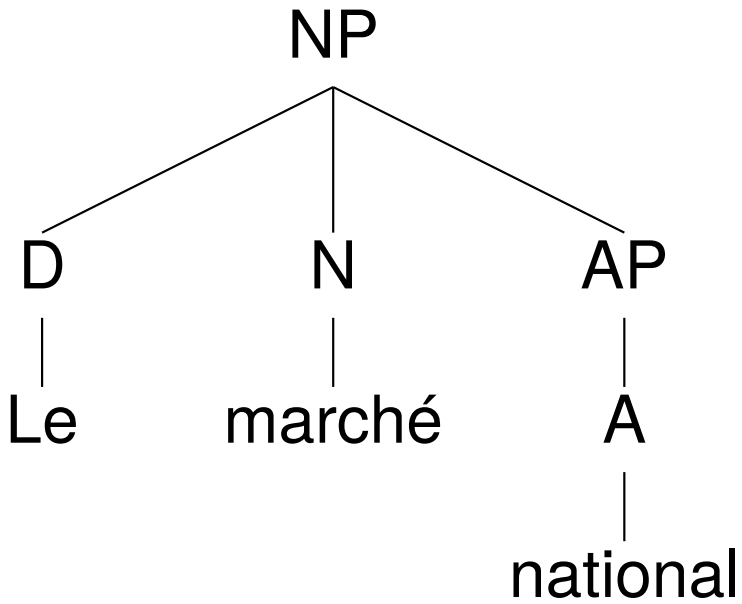
‘The national march’

Reference



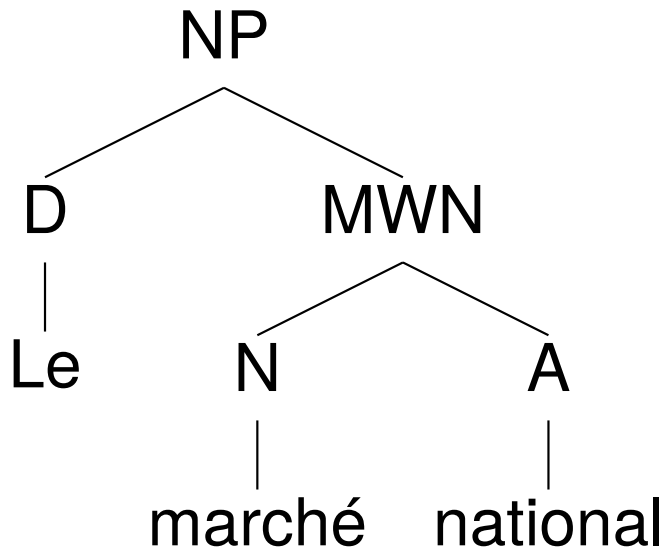
DP-TSG

DP-TSG errors: Overgeneration



‘The national march’

Reference



DP-TSG

MWEs are subtle; reference sometimes inconsistent

Standard Parsing Evaluation

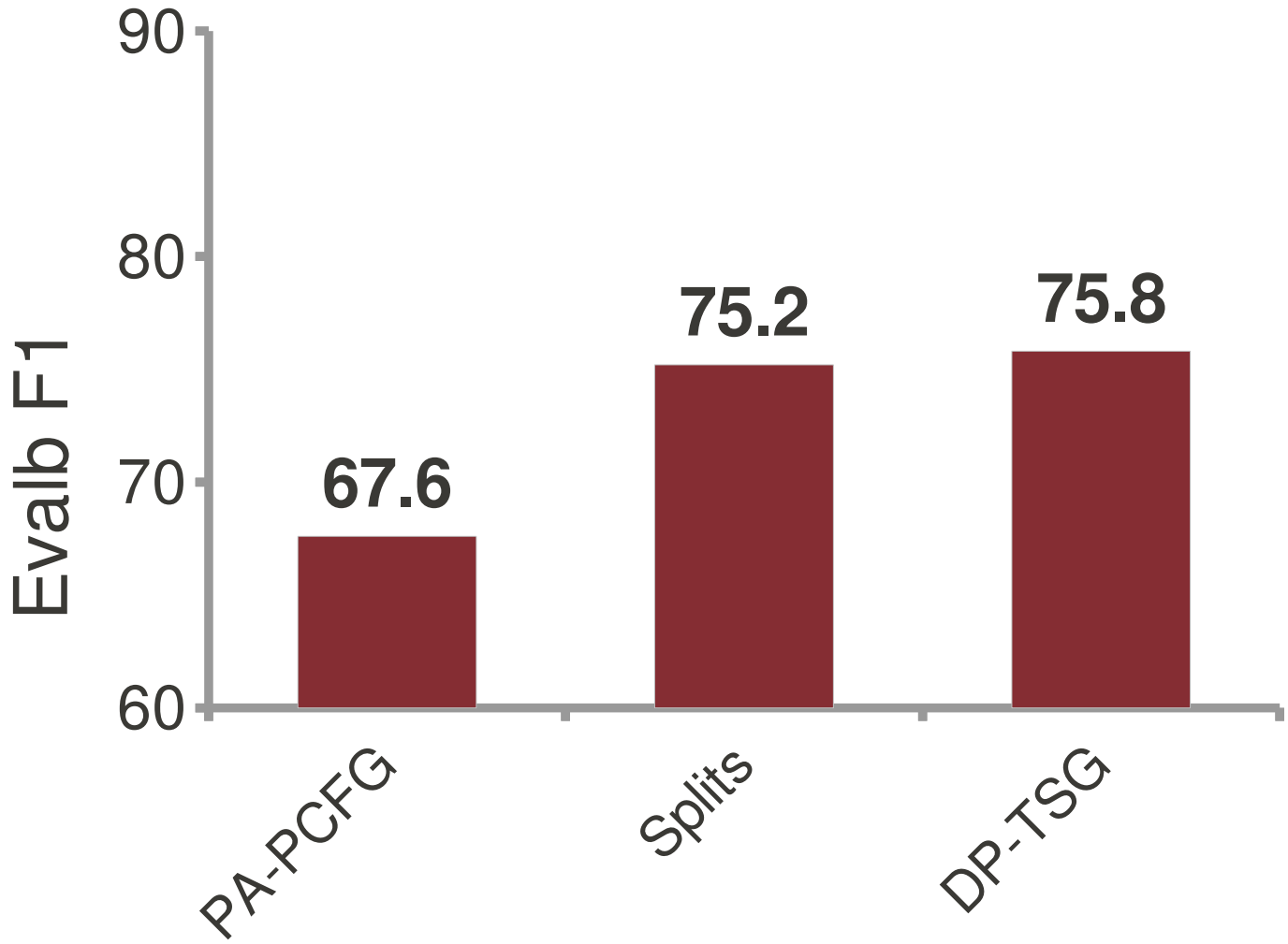
Same setup as MWE identification!

Standard Parsing Evaluation

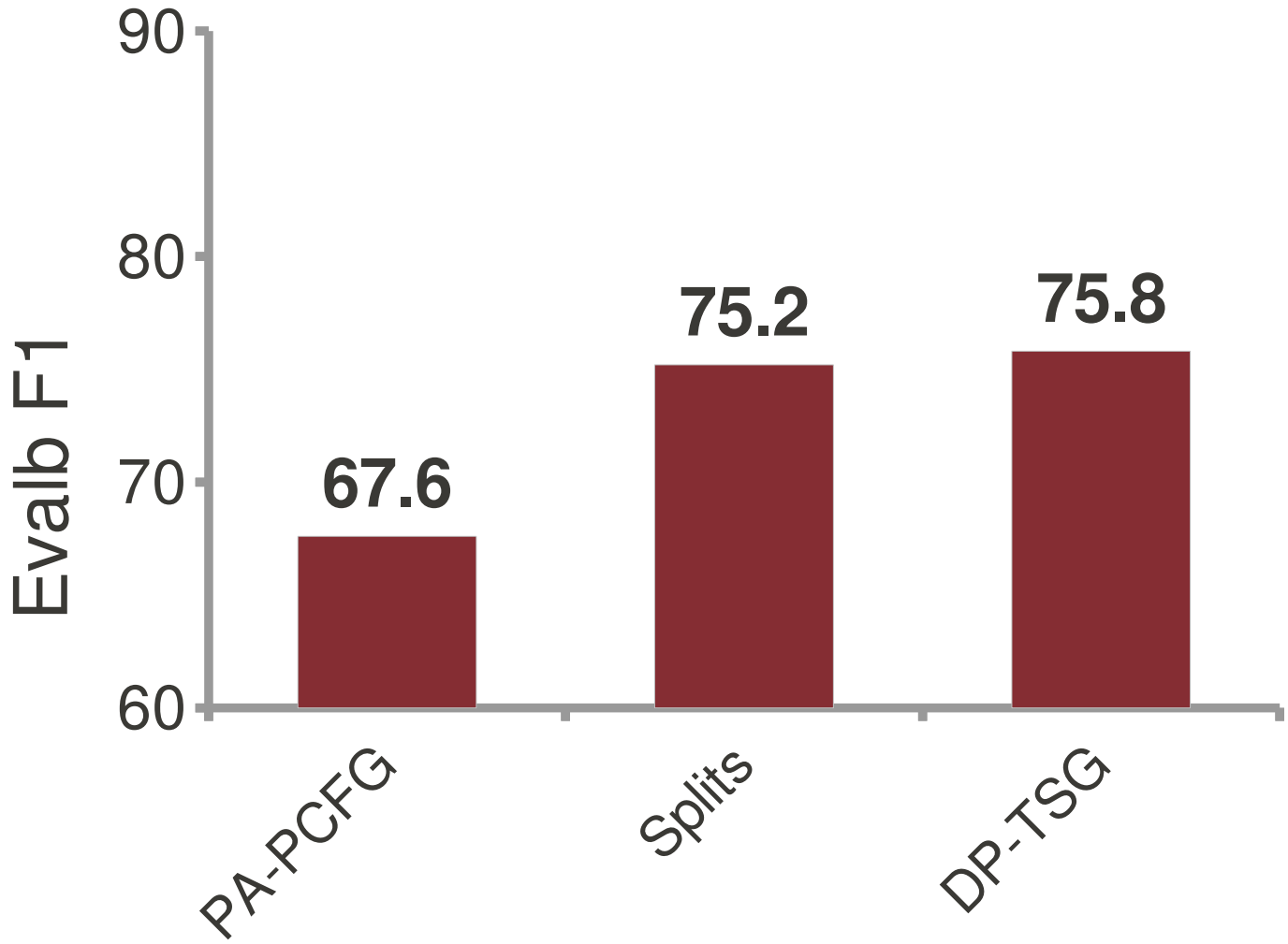
Same setup as MWE identification!

- Corpus: Paris 7 French Treebank (FTB)
- Split: same as (Crabbé and Candito 2008)
- Metrics: **Evalb** and Leaf Ancestor
- Lengths $\leq$ 40 words

French Parsing Evaluation: All bracketings



French Parsing Evaluation: All bracketings



Paper: more results (Stanford, Berkeley, etc.)

Future Directions

Syntactic context for *n*-gram methods

- Parse the corpus!
- Adapt lexical context measures to syntactic context

Future Directions

Syntactic context for *n*-gram methods

- Parse the corpus!
- Adapt lexical context measures to syntactic context

DP-TSG

- Better base distribution

Conclusion

Parsers work well for MWE identification

Conclusion

Parsers work well for MWE identification

Other languages: combine treebanks with MWE lists

Conclusion

Parsers work well for MWE identification

Other languages: combine treebanks with MWE lists

Non-“gold mode” parsing results for French

Conclusion

Parsers work well for MWE identification

Other languages: combine treebanks with MWE lists

Non-“gold mode” parsing results for French

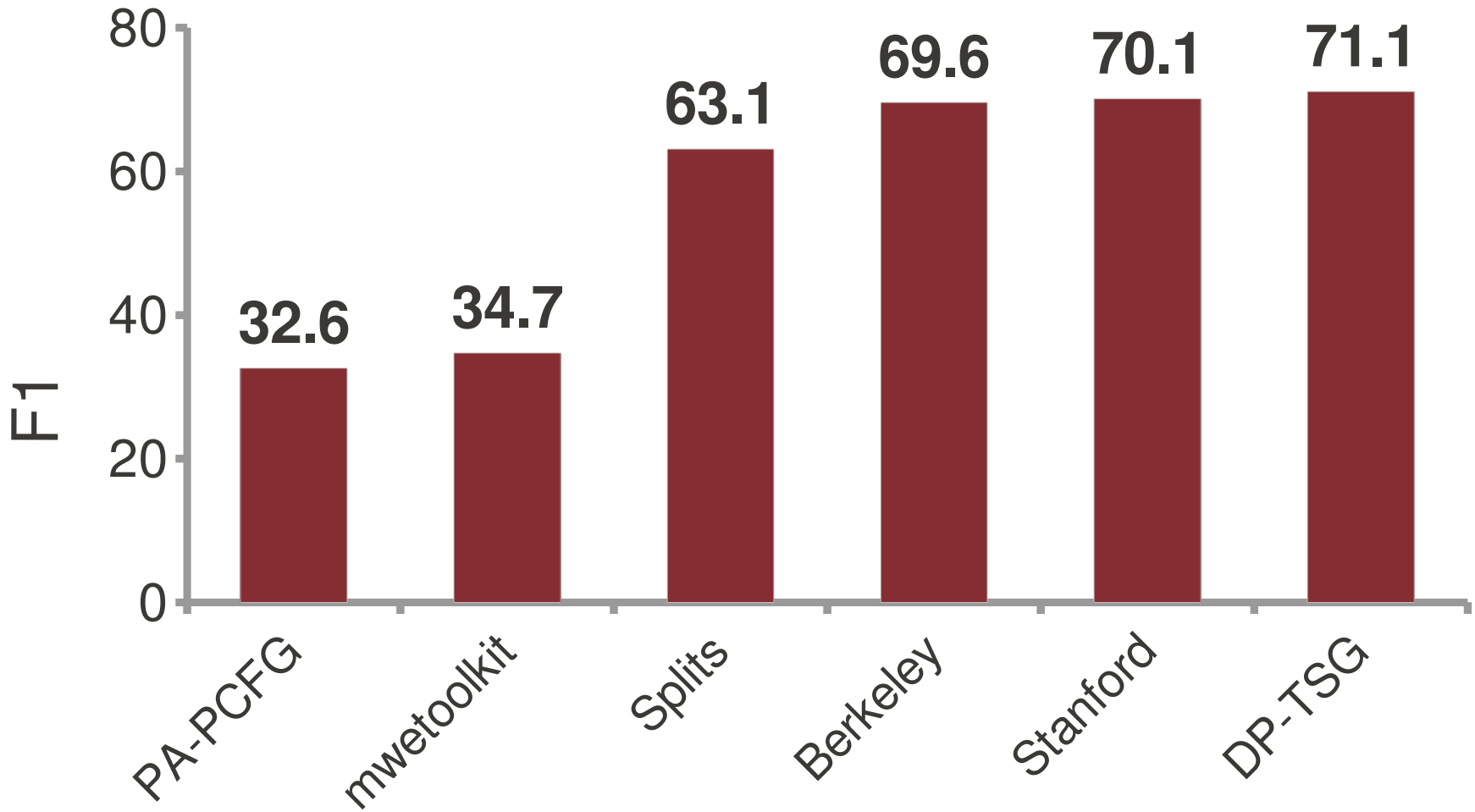
Code $\rightarrow$ Google: “Stanford parser”

un grand merci.

thanks a lot.

Questions?

MWE Identification Results



Dirichlet process TSG

DP prior for each non-terminal type $c \in V$:

$$\theta_c | c, \alpha_c, P_0(\cdot|c) \sim DP(\alpha_c, P_0)$$
$$e|\theta_c \sim \theta_c$$

[2]Cohn, Goldwater, and Blunsom 2009; Post and Gildea 2009; O'Donnell, Tenenbaum, and Goodman 2009.

Dirichlet process TSG

DP prior for each non-terminal type $c \in V$:

$$\theta_c | c, \alpha_c, P_0(\cdot|c) \sim DP(\alpha_c, P_0)$$
$$e | \theta_c \sim \theta_c$$

Binary variable b_s for each non-terminal node in corpus

- Supervised case: segment the treebank[2]

[2] Cohn, Goldwater, and Blunsom 2009; Post and Gildea 2009; O'Donnell, Tenenbaum, and Goodman 2009.

DP-TSG: Base distribution P_0

Phrasal rules:

$$P_0(A^+ \rightarrow B^- \; C^+) = p_{\text{MLE}}(A \rightarrow B \; C) \; s_B(1 - s_C)$$

DP-TSG: Base distribution P_0

Phrasal rules:

$$P_0(A^+ \rightarrow B^- \; C^+) = p_{\text{MLE}}(A \rightarrow B \; C) \; s_B(1 - s_C)$$

p_{MLE} is the manually-split grammar!

s_B is the *stop probability*

DP-TSG: Base distribution P_0

Lexical insertion rules:

$$P_0(C^+ \to t) = p_{\text{MLE}}(C \to t)\, p(t)$$

DP-TSG: Base distribution P_0

Lexical insertion rules:

$$P_0(C^+ \to t) = p_{\text{MLE}}(C \to t)\, p(t)$$

$p(t)$ is unigram probability of word t

Tree substitution grammars

A Probabilistic TSG is a 5-tuple $\langle V, \Sigma, R, \diamondsuit, \boldsymbol{\theta} \rangle$

- $c \in V$ are non-terminals
- $\diamondsuit \in V$ is a unique start symbol
- $t \in \Sigma$ are terminals
- $e \in R$ are **elementary trees**
- $\theta_{c,e} \in \boldsymbol{\theta}$ are parameters for each tree fragment

Tree substitution grammars

A Probabilistic TSG is a 5-tuple $\langle V, \Sigma, R, \diamondsuit, \theta \rangle$

- $c \in V$ are non-terminals
- $\diamondsuit \in V$ is a unique start symbol
- $t \in \Sigma$ are terminals
- $e \in R$ are **elementary trees**
- $\theta_{c,e} \in \theta$ are parameters for each tree fragment

elementary tree == tree fragment