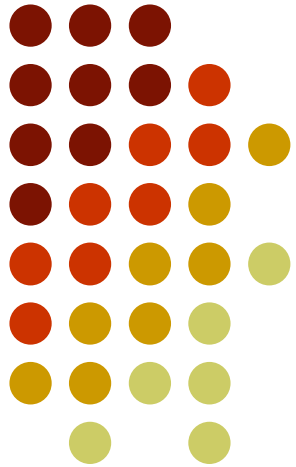
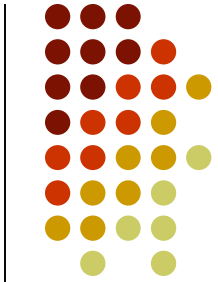
Markov Logic Networks: A Unified Approach To Language Processing

Pedro Domingos
Dept. of Computer Science & Eng.
University of Washington

Joint work with Stanley Kok, Daniel Lowd, Hoifung Poon, Matt Richardson, Parag Singla, Marc Sumner, and Jue Wang



Overview



- **Motivation**
- Background
- Markov logic
  - Inference
  - Learning
- Applications
  - Coreference resolution
- Discussion

Pipeline vs. Joint Architectures

- Most language processing systems have a pipeline architecture
- Simple, but errors accumulate
- We need joint inference across all stages
- Potentially much more accurate, but also much more complex

What We Need

- A common representation for all the stages
- A modeling language that enables this
- Efficient inference and learning algorithms
- Automatic compilation of model spec
- Makes language processing “plug and play”

Markov Logic

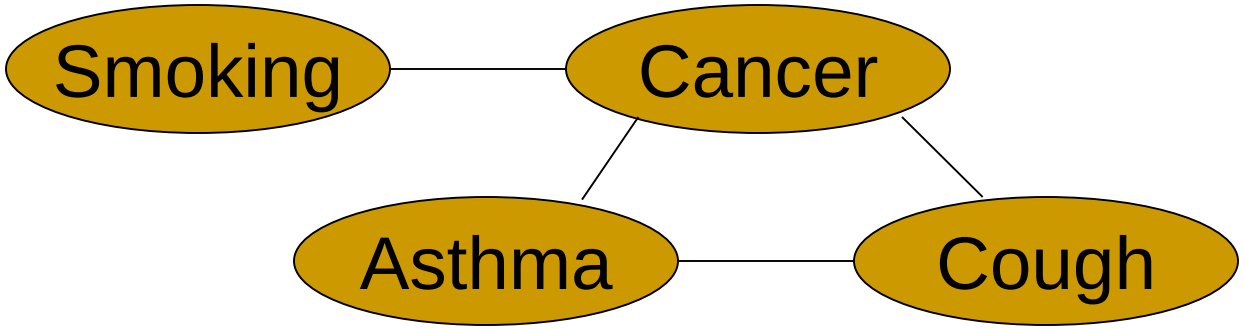
- **Syntax:** Weighted first-order formulas
- **Semantics:** Templates for Markov nets
- **Inference:** Lifted belief propagation
- **Learning:**
  - **Weights:** Convex optimization
  - **Formulas:** Inductive logic programming
- **Applications:** Coreference resolution, information extraction, semantic role labeling, ontology induction, etc.

Overview

- Motivation
- **Background**
- Markov logic
  - Inference
  - Learning
- Applications
  - Coreference resolution
- Discussion

Markov Networks

- **Undirected** graphical models



- Potential functions defined over cliques

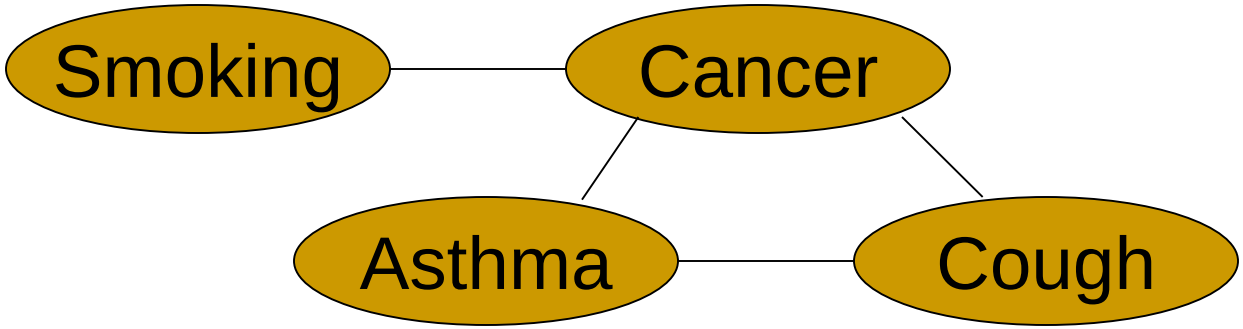
$$P(x) = \frac{1}{Z} \prod_c \Phi_c(x_c)$$

$$Z = \sum_x \prod_c \Phi_c(x_c)$$

Smoking	Cancer	Φ(S,C)
False	False	4.5
False	True	4.5
True	False	2.7
True	True	4.5

Markov Networks

- **Undirected** graphical models



- Log-linear model:

$$P(x) = \frac{1}{Z} \exp\left(\sum_i w_i f_i(x) \right)$$

Weight of Feature *i*

Feature *i*

$$f_1(\text{Smoking}, \quad \text{Cancer}) = \begin{cases} 1 & \text{if} \quad \neg \text{Smoking} \quad \vee \quad \text{Cancer} \\ 0 & \text{otherwise} \end{cases}$$

$$w_1 = 1.5$$

First-Order Logic

- **Symbols:** Constants, variables, functions, predicates
  E.g.: Anna, x, MotherOf(x), Friends(x, y)
- **Logical connectives:** Conjunction, disjunction, negation, implication, quantification, etc.
- **Grounding:** Replace all variables by constants
  E.g.: Friends (Anna, Bob)
- **World**: Assignment of truth values to all ground atoms

Example: Heads and Appositions

Mentions of Bush are often headed by "Bush"
Mentions of Bush are often headed by "President"
Appositions usually refer to the same entity

Example: Heads and Appositions

$$\forall x\ \text{MentionOf}(x, \text{Bush}) \Rightarrow \text{Head}(x, \text{"Bush"})$$

$$\forall x\ \text{MentionOf}(x, \text{Bush}) \Rightarrow \text{Head}(x, \text{"President"})$$

$$\forall x, y, c\ \text{Apposition}(x, y) \wedge \text{MentionOf}(x, c) \Rightarrow \text{MentionOf}(y, c)$$

Overview

- Motivation
- Background
- **Markov logic**
  - Inference
  - Learning
- Applications
  - Coreference resolution
- Discussion

Markov Logic

- A logical KB is a set of **hard constraints** on the set of possible worlds
- Let's make them **soft constraints**:
  When a world violates a formula,
  It becomes less probable, not impossible
- Give each formula a **weight**
  (Higher weight $\Rightarrow$ Stronger constraint)

$$P(\text{world}) \propto \exp\left(\sum \text{weights of formulas it satisfies}\right)$$

Definition

- A Markov Logic Network (MLN) is a set of pairs (F, w) where
  - F is a formula in first-order logic
  - w is a real number
- Together with a set of constants, it defines a Markov network with
  - One node for each grounding of each predicate in the MLN
  - One feature for each grounding of each formula F in the MLN, with the corresponding weight w

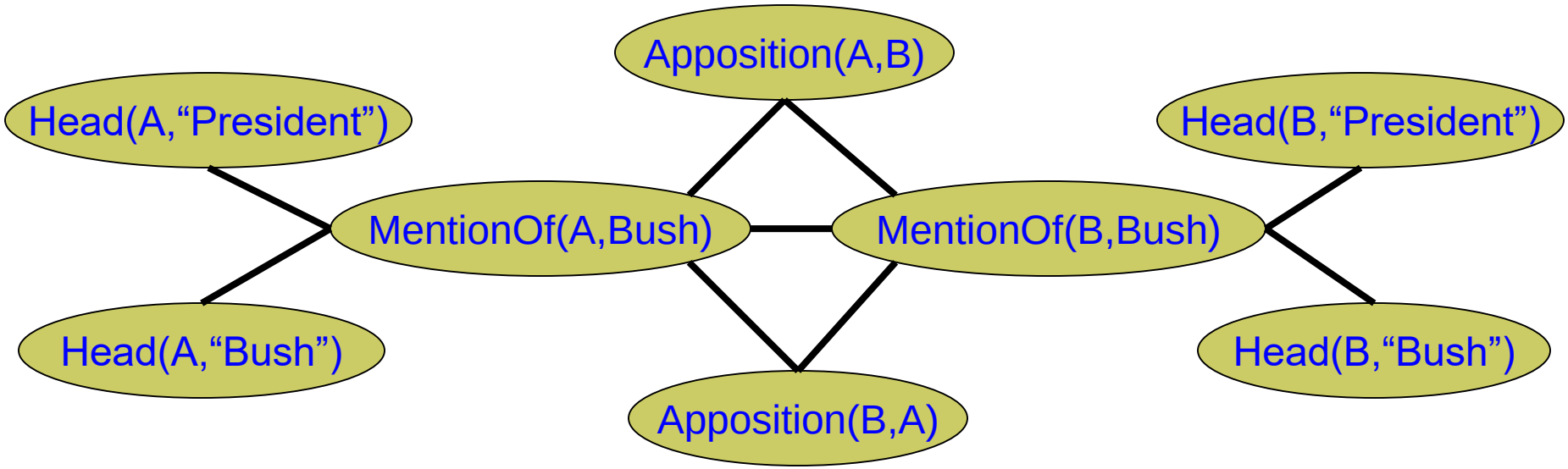
Example: Heads and Appositions

1.5	$\forall x\ \text{MentionOf}(x, \text{Bush}) \Rightarrow \text{Head}(x, \text{"Bush"})$
0.8	$\forall x\ \text{MentionOf}(x, \text{Bush}) \Rightarrow \text{Head}(x, \text{"President"})$
100	$\forall x, y, c\ \text{Apposition}(x, y) \wedge \text{MentionOf}(x, c) \Rightarrow \text{MentionOf}(y, c)$

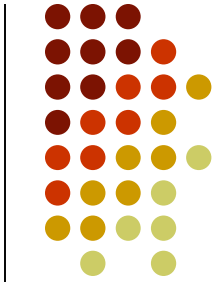
Example: Heads and Appositions

1.5	$\forall x\ \text{MentionOf}(x, \text{Bush}) \Rightarrow \text{Head}(x, \text{"Bush"})$
0.8	$\forall x\ \text{MentionOf}(x, \text{Bush}) \Rightarrow \text{Head}(x, \text{"President"})$
100	$\forall x, y, c\ \text{Apposition}(x, y) \wedge \text{MentionOf}(x, c) \Rightarrow \text{MentionOf}(y, c)$

Two mention constants: **A** and **B**



Markov Logic Networks



- MLN is **template** for ground Markov nets
- Probability of a world x:

$$P(x) = \frac{1}{Z} \exp\left(\sum_i w_i n_i(x) \right)$$

Weight of formula *i*

No. of true groundings of formula *i* in *x*

- **Typed** variables and constants greatly reduce size of ground Markov net
- Functions, existential quantifiers, etc.
- Infinite and continuous domains

Relation to Statistical Models

- Special cases:
  - Markov networks
  - Markov random fields
  - Bayesian networks
  - Log-linear models
  - Exponential models
  - Max. entropy models
  - Gibbs distributions
  - Boltzmann machines
  - Logistic regression
  - Hidden Markov models
  - Conditional random fields
- Obtained by making all predicates zero-arity
- Markov logic allows objects to be interdependent (non-i.i.d.)

Relation to First-Order Logic

- Infinite weights $\Rightarrow$ First-order logic
- Satisfiable KB, positive weights $\Rightarrow$ Satisfying assignments = Modes of distribution
- Markov logic allows contradictions between formulas

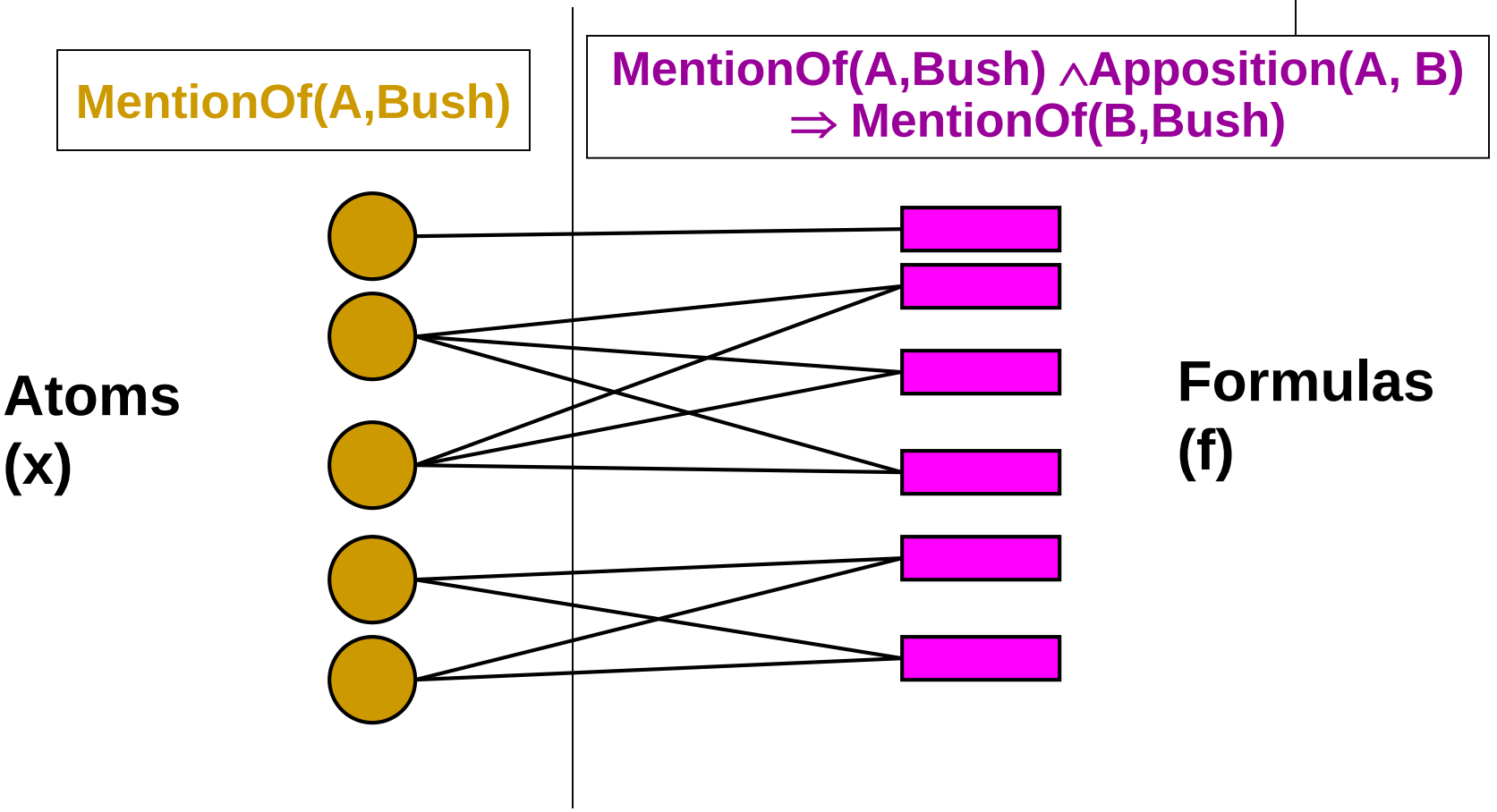
Overview

- Motivation
- Background
- Markov logic
  - **Inference**
  - Learning
- Applications
  - Coreference resolution
- Discussion

Belief Propagation

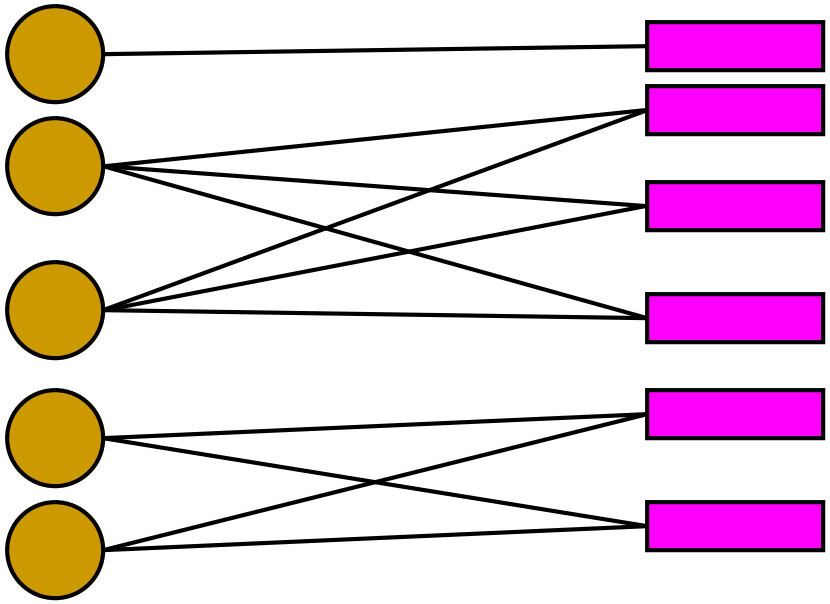
- Goal: Compute probabilities or MAP state
- Belief propagation: Subsumes Viterbi, etc.
- Bipartite network
  - Variables = Ground atoms
  - Features = Ground formulas
- Repeat until convergence:
  - Nodes send messages to their features
  - Features send messages to their variables
- Messages = Approximate marginals

Belief Propagation



Belief Propagation

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$

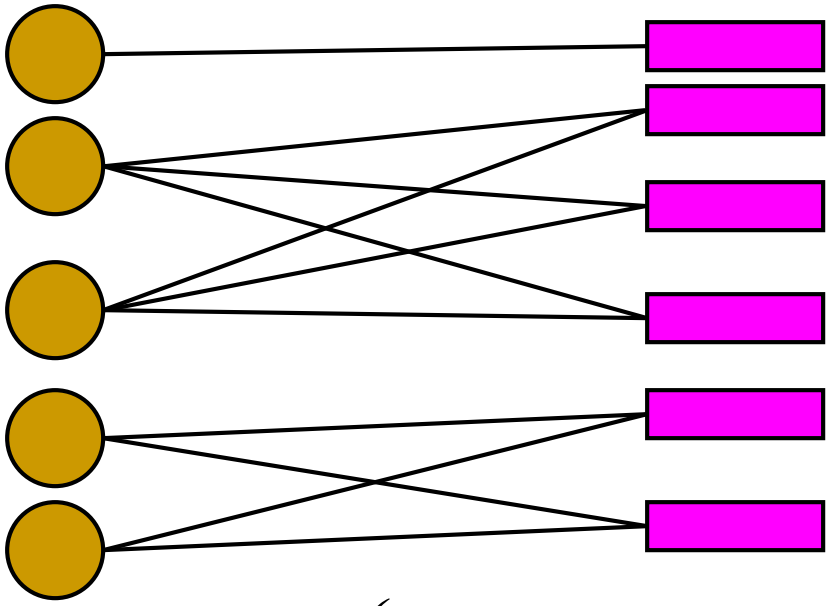


Atoms (x)

Formulas (f)

Belief Propagation

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$



Atoms (x)

Formulas (f)

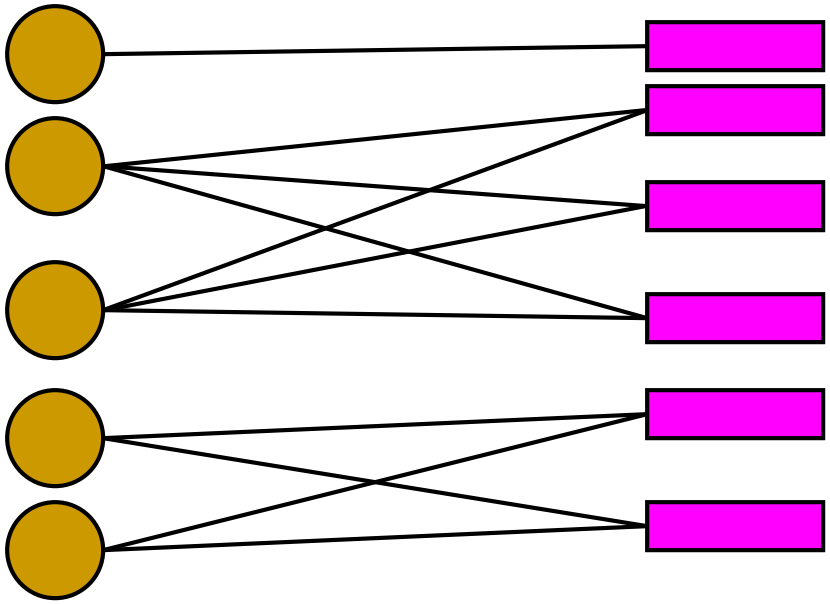
$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left(e^{wf(x)} \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

But This Is Too Slow

- One message for each atom/formula pair
- Can easily have billions of formulas
- Too many messages!
- Group atoms/formulas which pass same message (as in resolution)
- One message for each pair of clusters
- Greatly reduces the size of the network

Belief Propagation

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$



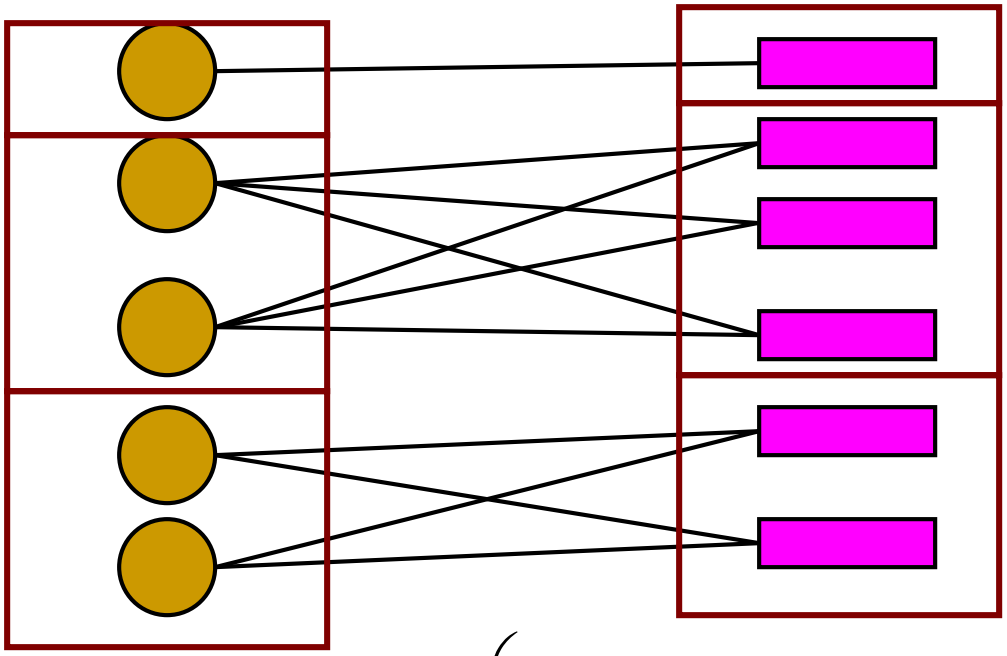
Atoms (x)

Formulas (f)

$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left(e^{wf(x)} \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

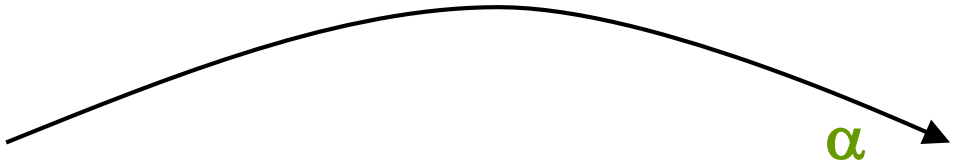
Lifted Belief Propagation

$$\mu_{x \to f}(x) = \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$

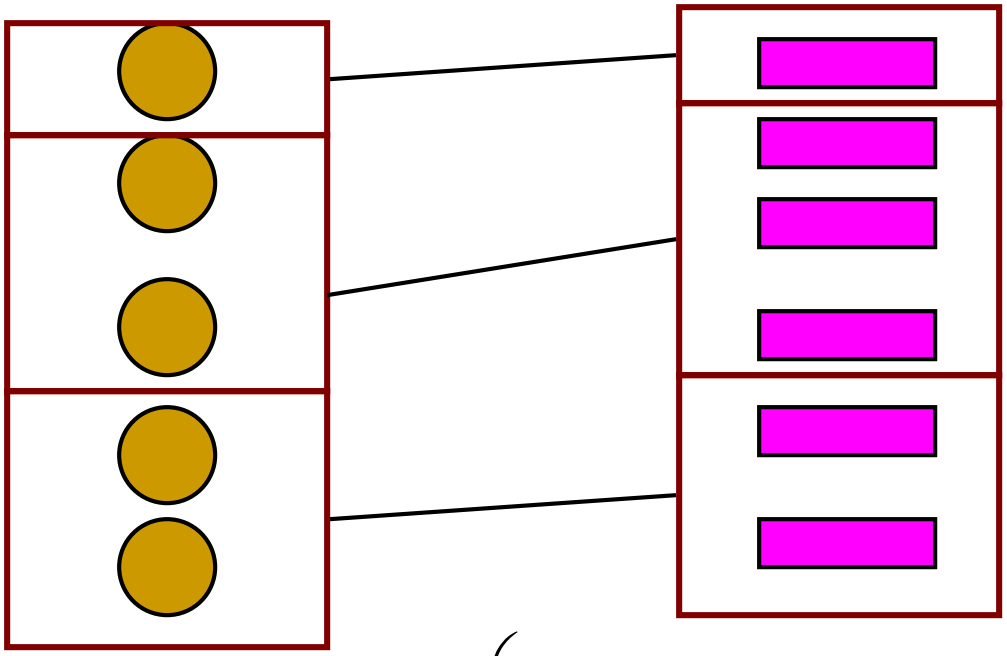


$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left(e^{wf(x)} \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

Lifted Belief Propagation



$$\mu_{x \to f}(x) = \beta \prod_{h \in n(x) \setminus \{f\}} \mu_{h \to x}(x)$$



$$\mu_{f \to x}(x) = \sum_{\sim\{x\}} \left(e^{wf(x)} \prod_{y \in n(f) \setminus \{x\}} \mu_{y \to f}(y) \right)$$

Lifted Belief Propagation

- Form **lifted network**
  - **Supernode:** Set of ground atoms that all send and receive same messages throughout BP
  - **Superfeature:** Set of ground clauses that all send and receive same messages throughout BP
- Run belief propagation on lifted network
- Same results as ground BP
- Time and memory savings can be huge

Forming the Lifted Network

1. **Form initial supernodes**
   One per predicate and truth value
   (true, false, unknown)
2. **Form superfeatures by doing joins of their supernodes**
3. **Form supernodes by projecting superfeatures down to their predicates**
   Supernode = Groundings of a predicate with same number of projections from each superfeature
4. **Repeat until convergence**

Overview

- Motivation
- Background
- Markov logic
  - Inference
  - **Learning**
- Applications
  - Coreference resolution
- Discussion

Learning

- Data is a relational database
- Learning parameters (weights)
  - Supervised
  - Unsupervised
- Learning structure (formulas)

Supervised Learning

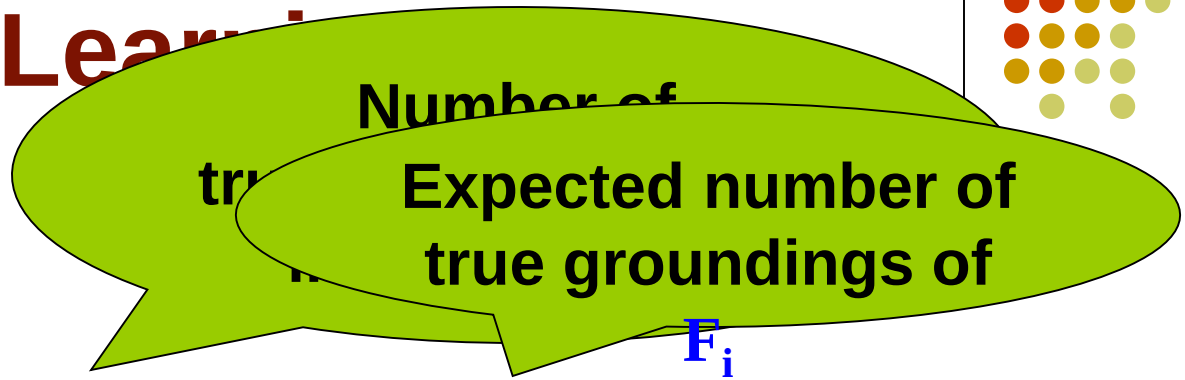
- Maximizes conditional log-likelihood:

$$L(x, y) = \log P(Y = y \mid X = x)$$

- $\boldsymbol{Y}$: Query variables
- $\boldsymbol{X}$: Evidence variables
- $\boldsymbol{x}$, $\boldsymbol{y}$: Observed values in training data

Supervised Learni

- Gradient:



$$\frac{\partial L(x,y)}{\partial w_i} = \mathrm{N}_i(x,y) - \mathrm{E}_{Y \mid x}[\mathrm{N}_i]$$

- Use inference to compute $\mathrm{E}[\mathrm{N}_i]$
- Preconditioned scaled conjugate gradient (PSCG) [Lowd & Domingos, 2007]

Unsupervised Learning

- Maximizes marginal cond. log-likelihood:

$$L(x, y) = \log \sum_{z} P(Y = y, Z = z \mid X = x)$$

- $\boldsymbol{Y}$: Query variables
- $\boldsymbol{X}$: Evidence variables
- $\boldsymbol{x}$, $\boldsymbol{y}$: Observed values in the training data
- $\boldsymbol{Z}$: Hidden variables

Unsupervised Learning

- Gradient:

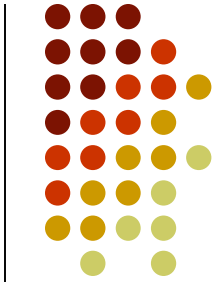
$$\frac{\partial L(x,y)}{\partial w_i} = \mathrm{E}_{Z \mid x,y}[N_i] - \mathrm{E}_{Y,Z \mid x}[N_i]$$

- Use inference to compute both $\mathrm{E}[N_i]$s
- Also works for semi-supervised learning

Structure Learning

- Generalizes feature induction in Markov nets
- Any inductive logic programming approach can be used, but . . .
- Goal is to induce any clauses, not just Horn
- Evaluation function should be likelihood
- Requires learning weights for each candidate
- Turns out not to be bottleneck
- Bottleneck is counting clause groundings
- Solution: Subsampling

Overview



- Motivation
- Background
- Markov logic
  - Inference
  - Learning
- **Applications**
  - **Coreference resolution**
- Discussion

Applications

NLP

- Information extraction
- Coreference resolution
- Citation matching
- Semantic role labeling
- Ontology induction
- Etc.

Others

- Social network analysis
- Robot mapping
- Computational biology
- Probabilistic Cyc
- CALO
- Etc.

Coreference Resolution

- Identifies noun phrases (mentions) that refer to the same entity
- Can be viewed as clustering the mentions (each entity is a cluster)
- Key component in NLP applications

State of the Art

- Supervised learning
  - Classification (e.g., are two mentions coreferent?)
  - Requires expensive labeling
- Unsupervised learning
  - Still lags supervised approaches by a large margin
  - E.g., Haghighi & Klein [2007]
    - Most sophisticated to date
    - Lags supervised methods by as much as 7% F1
    - Generative model $\Rightarrow$ Nontrivial to extend with arbitrary dependencies

This Talk

First unsupervised coreference resolution system that rivals supervised approaches

MLNs for Coreference Resolution

- **Goal:** Infer the truth values of MentionOf(m, e) for every mention m and entity e
- Base MLN
- Joint inference
  - Appositions
  - Predicate nominals
- Full MLN = Base + Joint Inference
- Rule-based model

Base MLN: Formulas

- **N**
  E
- **P**
  E
- E
  E
- M
  number, and gender

ush”

der

ype,

9 predicates

17 formulas

No. weights = O(No. entities)

Base MLN: Exponential Priors

- Prior on total number of entities:

  $\text{weight} = -1$ (per entity)

- Prior on distance between each pronoun and its closest antecedent:

  $\text{weight} = -1$ (per pronominal mention)

Joint Inference

- Appositions

  E.g., "Mr. Bush, the President of the U.S.A., …"

- Predicate nominals

  E.g., "Mr. Bush ***is*** the President of the U.S.A."

- **Joint inference:**

  Mentions that are appositions or predicate nominals usually refer to the same entity

Rule-Based Model

- Cluster non-pronouns with same heads
- Place each pronoun in the entity with
  - The closest antecedent
  - No known conflicts in type, number, gender
- Can be encoded in MLN with just four formulas
- No learning
- Suffices to outperform Haghighi & Klein [2007]

Unsupervised Learning

- Maximizes marginal cond. log-likelihood:

$$L(x, y) = \log \sum_{z} P(Y = y, Z = z \mid X = x)$$

- $\boldsymbol{Y}$: Query variables
- $\boldsymbol{X}$: Evidence variables
- $\boldsymbol{x}$, $\boldsymbol{y}$: Observed values in the training data
- $\boldsymbol{Z}$: Hidden variables

Unsupervised Learning for Coreference Resolution MLNs

- ***Y*:** Heads, known properties
- ***X*:** Pronoun, apposition, predicate nominal
- ***Z*:** Coreference assignment (MentionOf), unknown properties

Evaluation

- Datasets
- Metrics
- Systems
- Results
- Analysis

Datasets

- MUC-6
- ACE-2004 training corpus
- ACE Phase II (ACE-2)

Metrics

- Precision, recall, F1 (MUC, B^3, Pairwise)
- Mean absolute error in number of entities

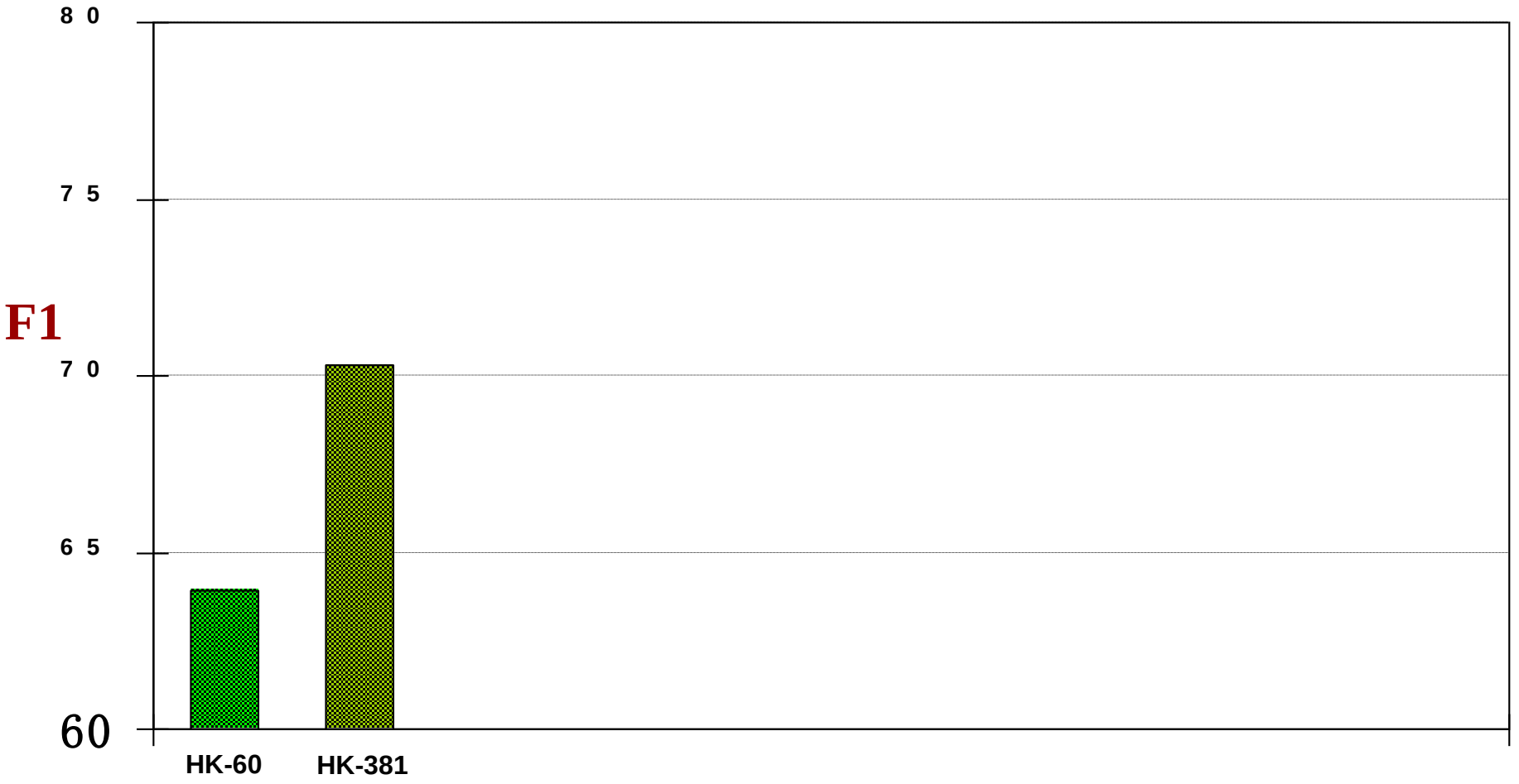
Systems: Recent Approaches

- **Unsupervised:** Haghighi & Klein [2007]
- **Supervised:**
  - McCallum & Wellner [2005]
  - Ng [2005]
  - Denis & Baldridge [2007]

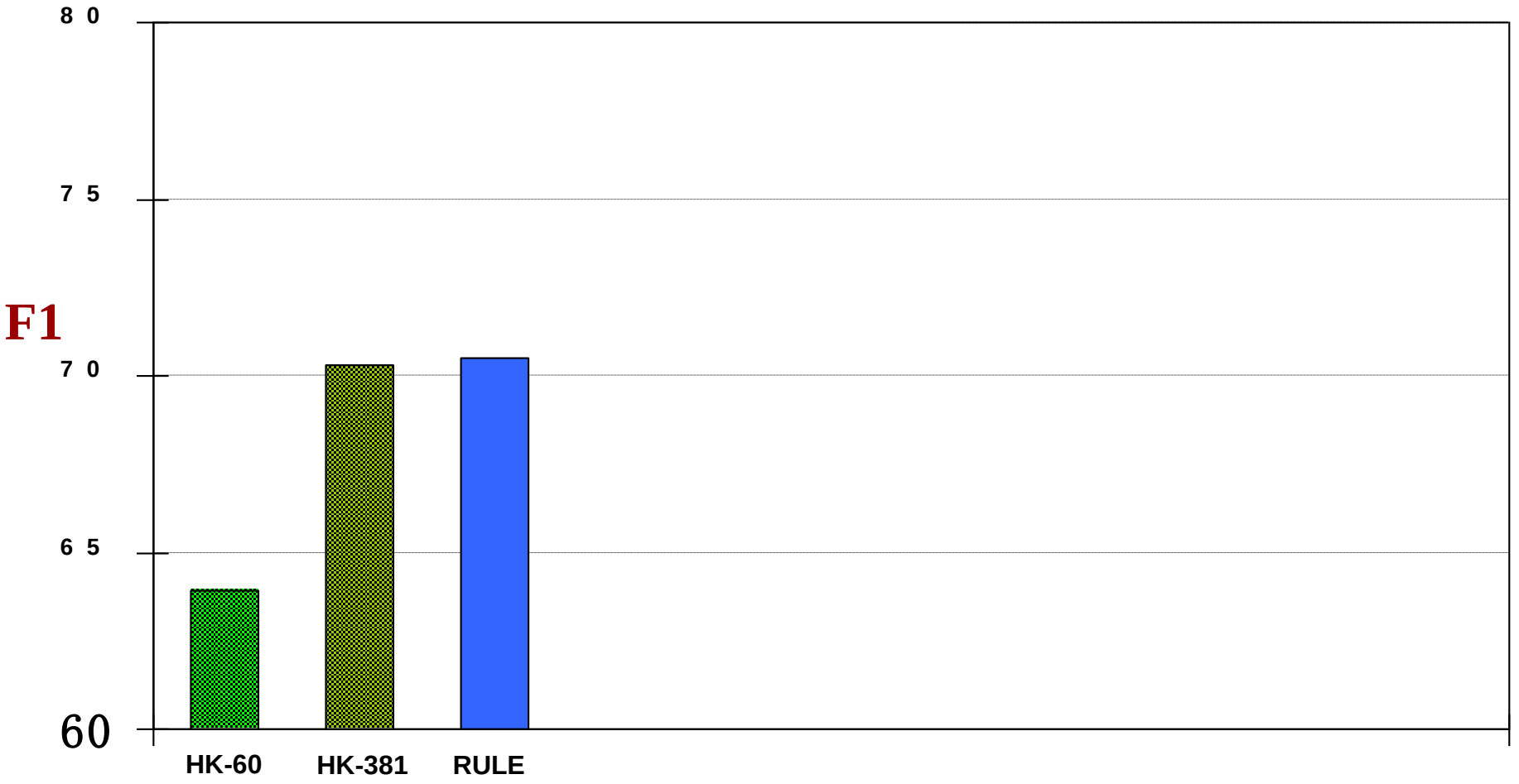
Systems: MLNs

- Rule-based model (**RULE**)
- Base MLN
  - **MLN-1:** trained on each document itself
  - **MLN-30:** trained on 30 test documents together
- Better head determination (**-H**)
- Joint inference with appositions (**-A**)
- Joint inference with predicate nominals (**-N**)

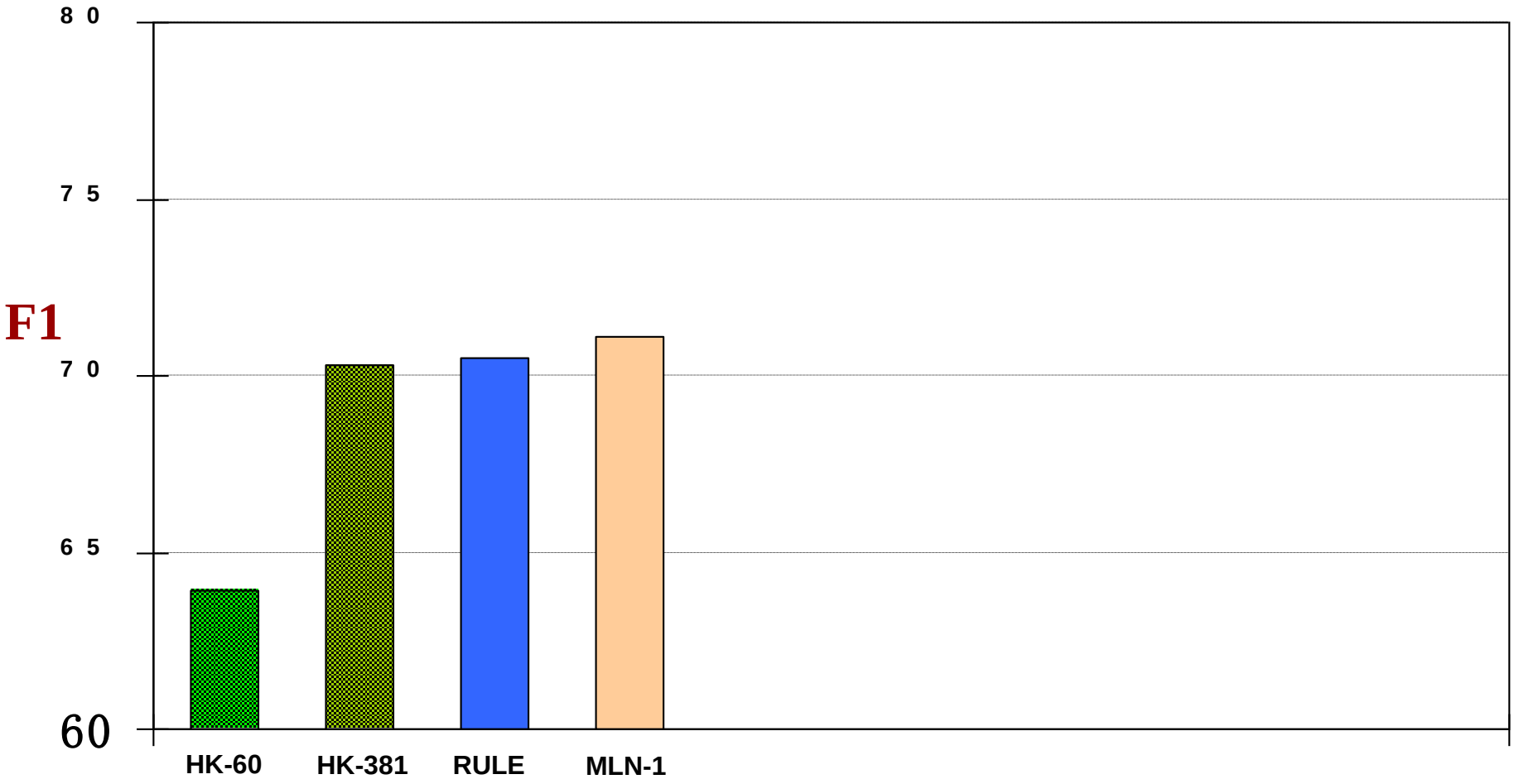
Results: MUC-6



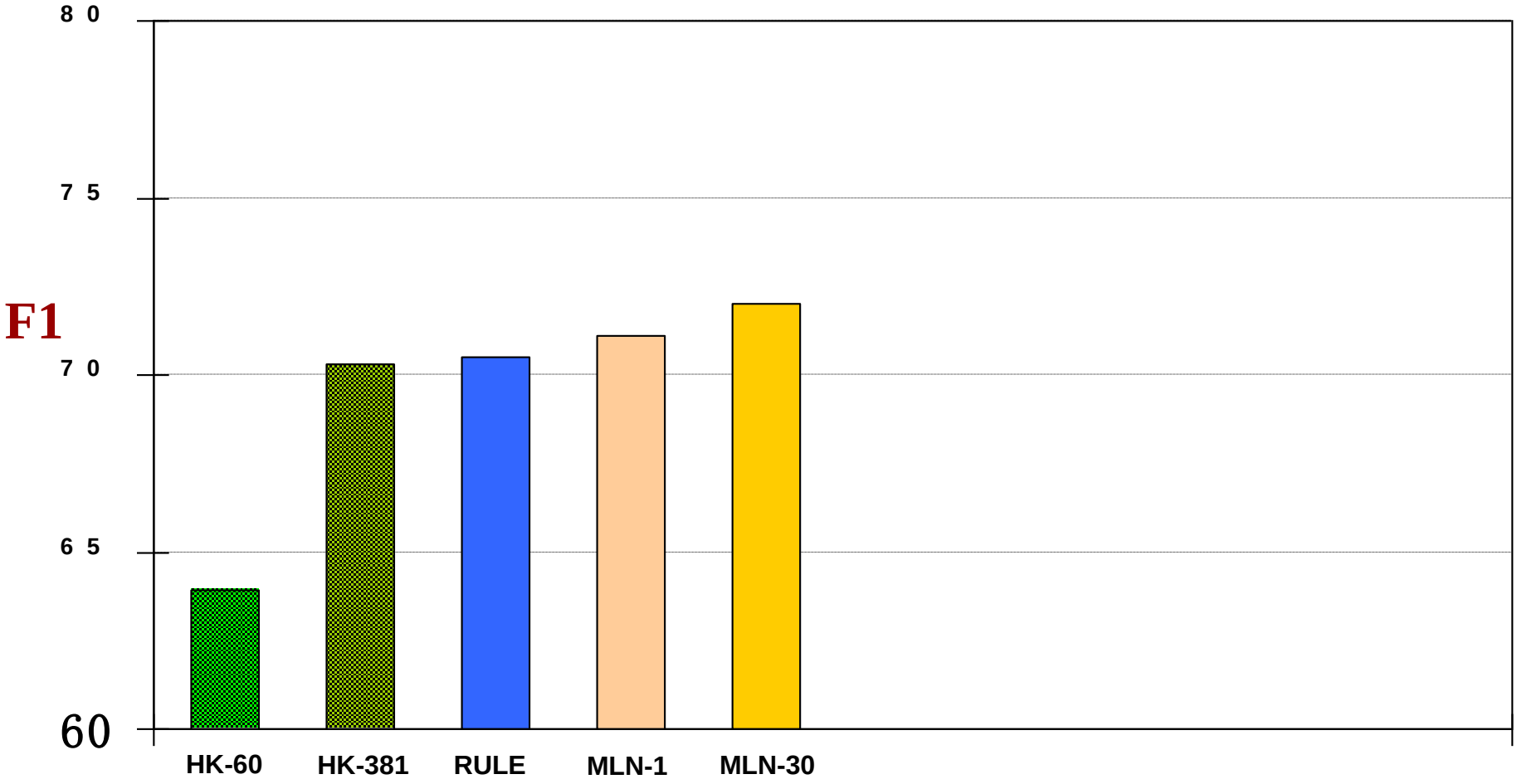
Results: MUC-6



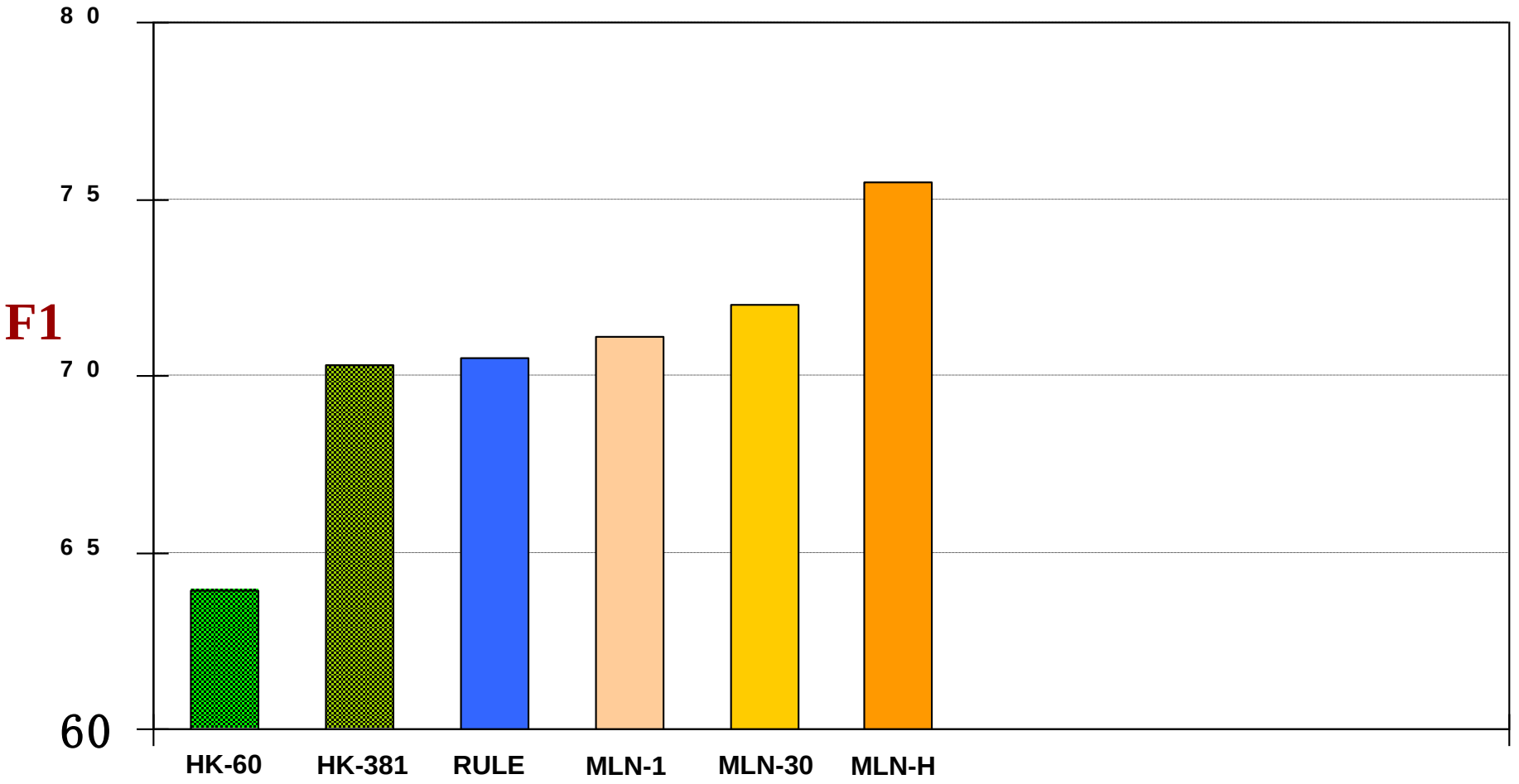
Results: MUC-6



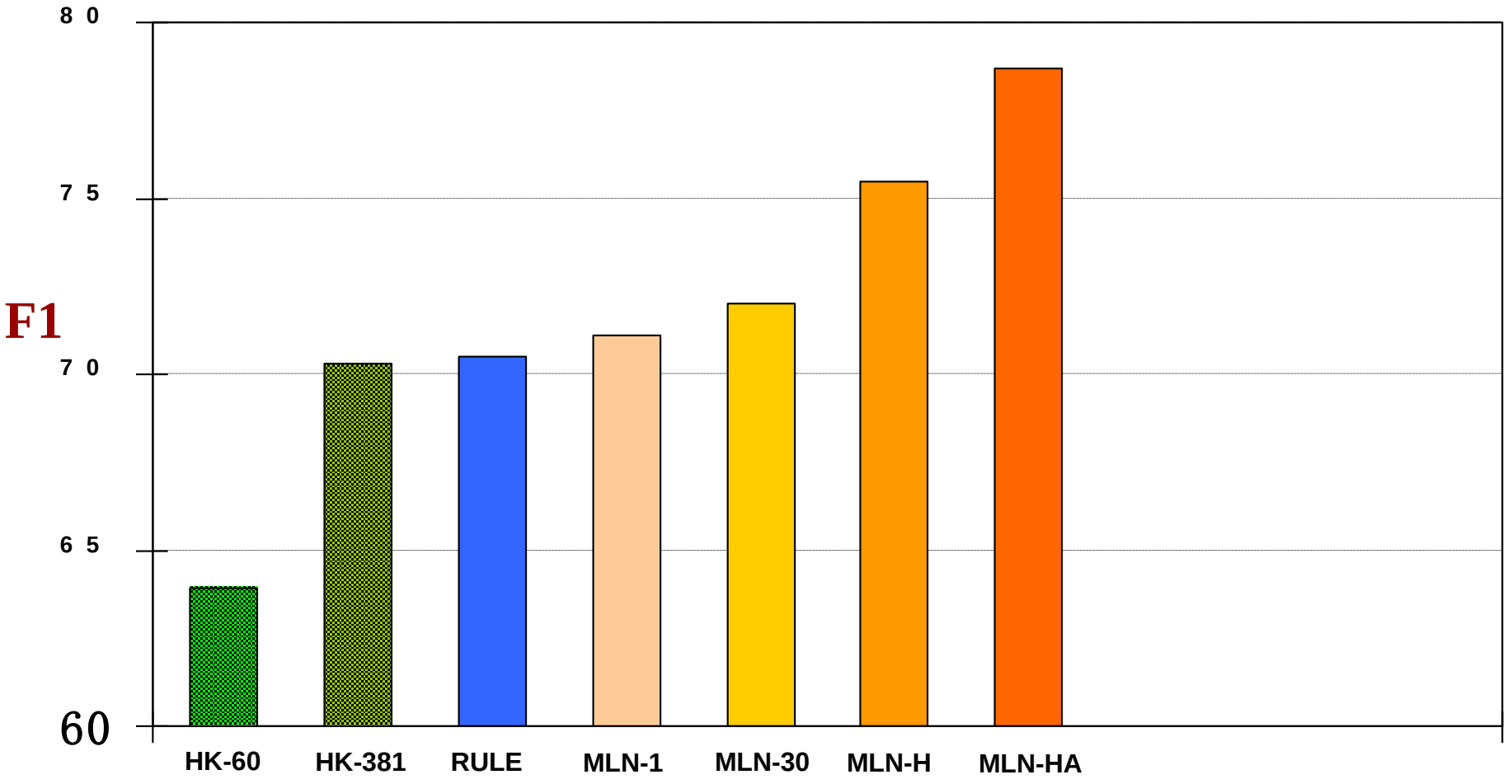
Results: MUC-6



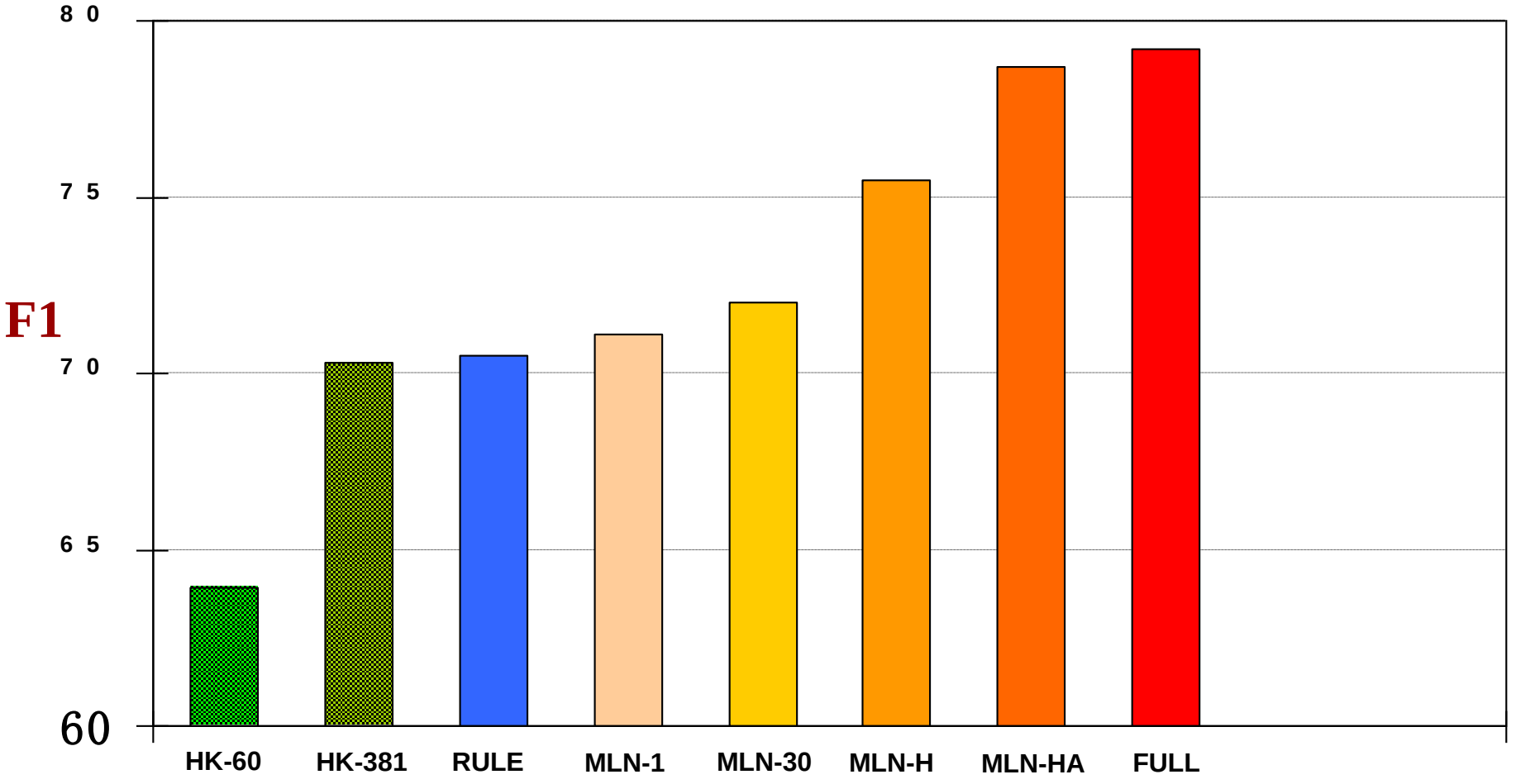
Results: MUC-6



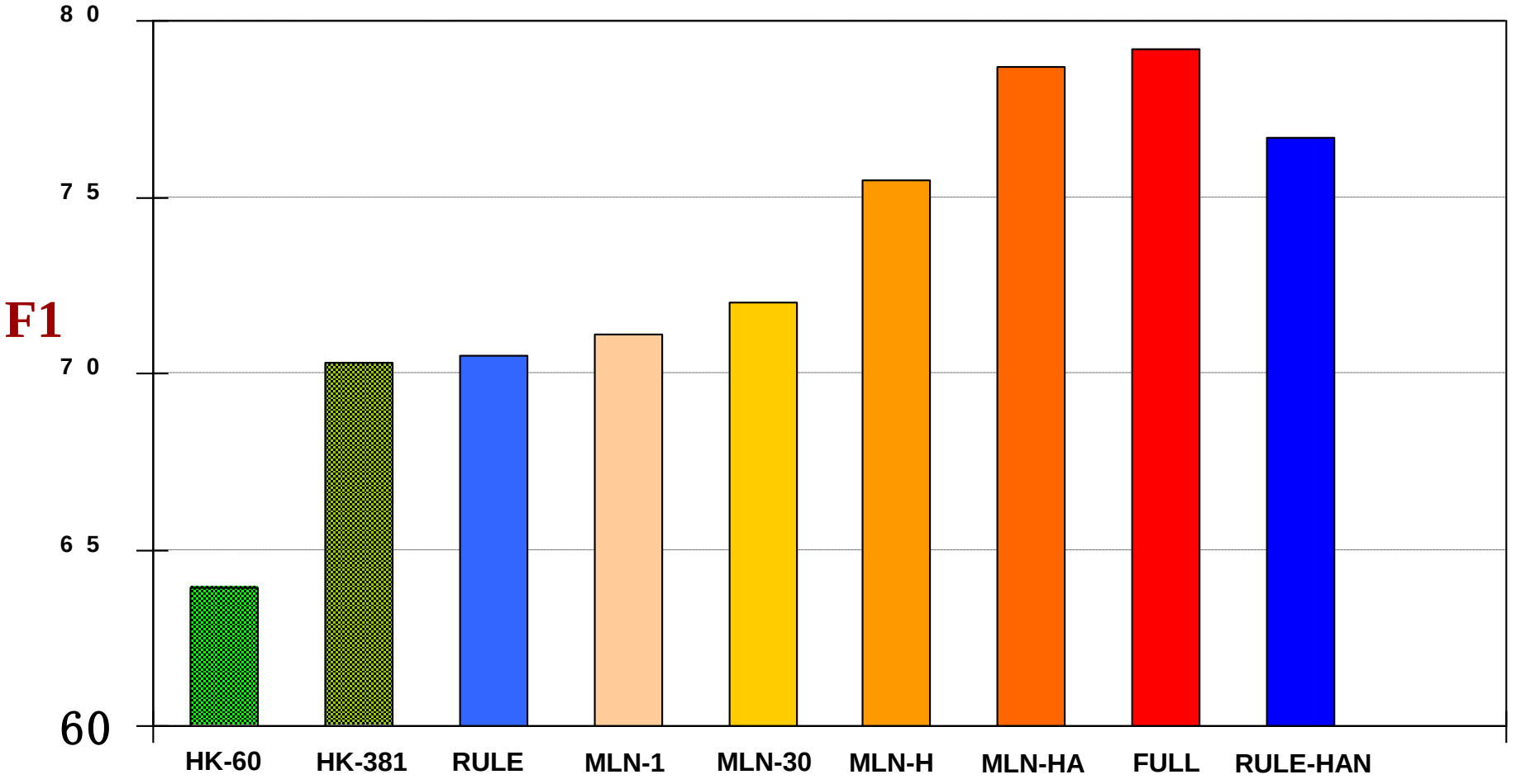
Results: MUC-6



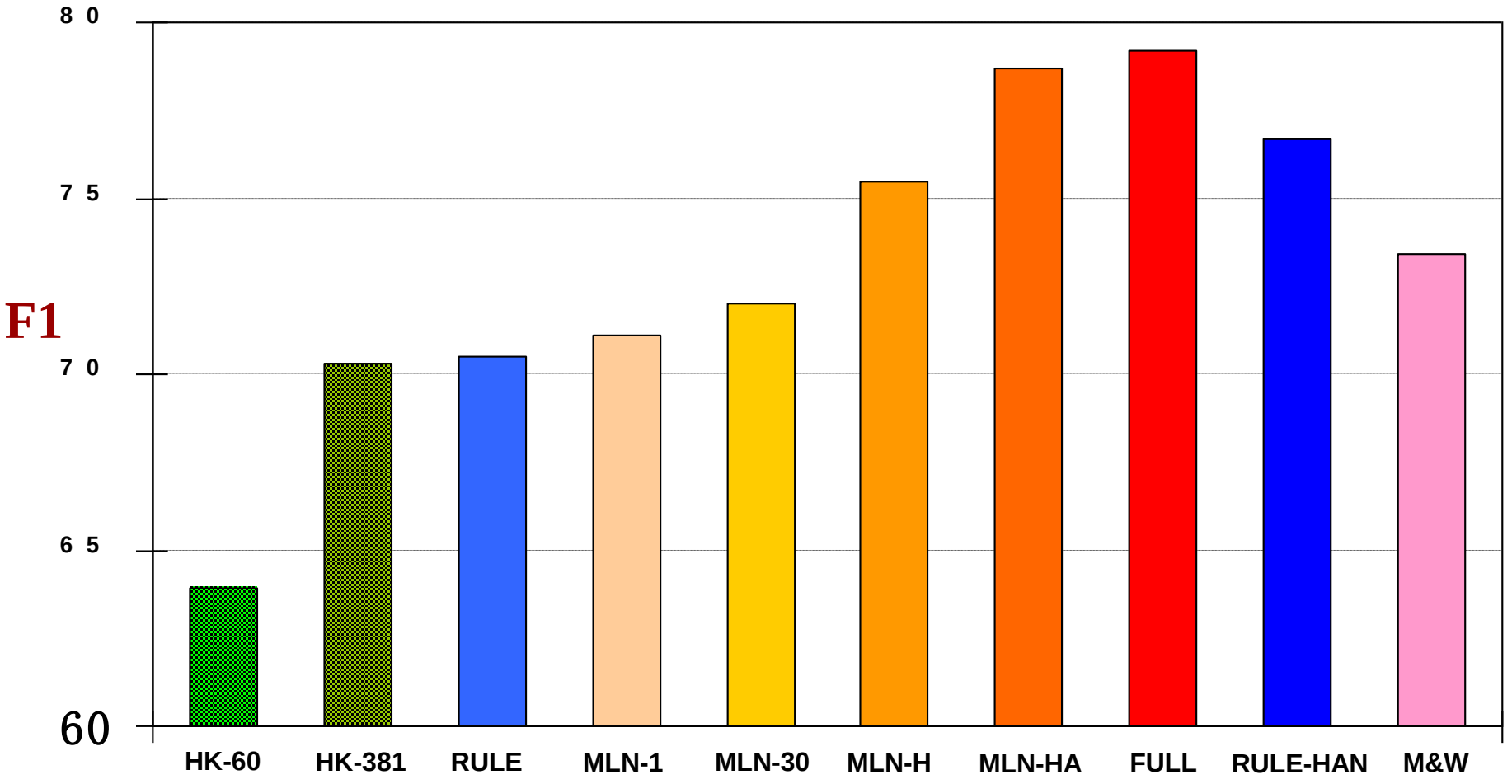
Results: MUC-6



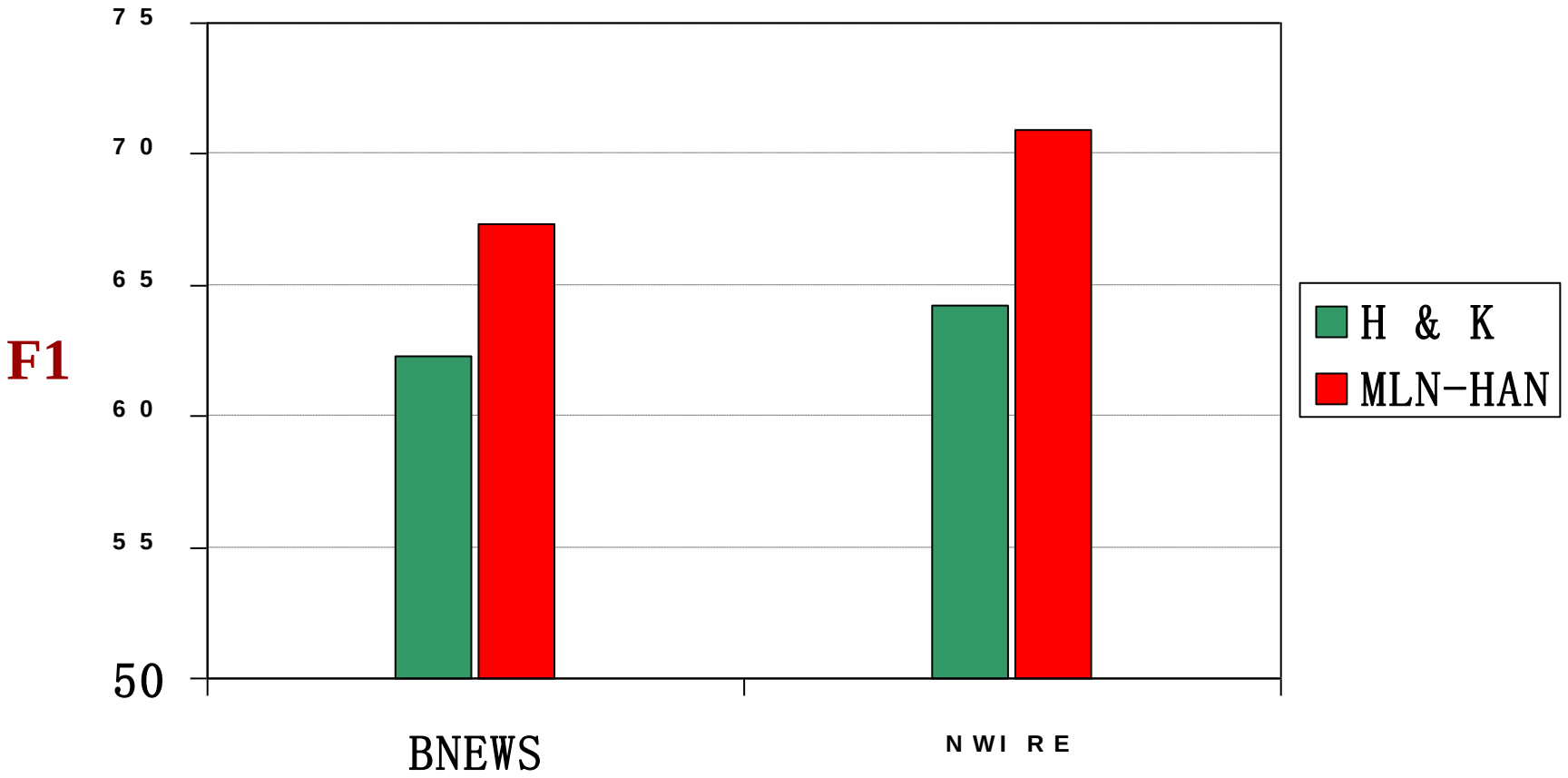
Results: MUC-6



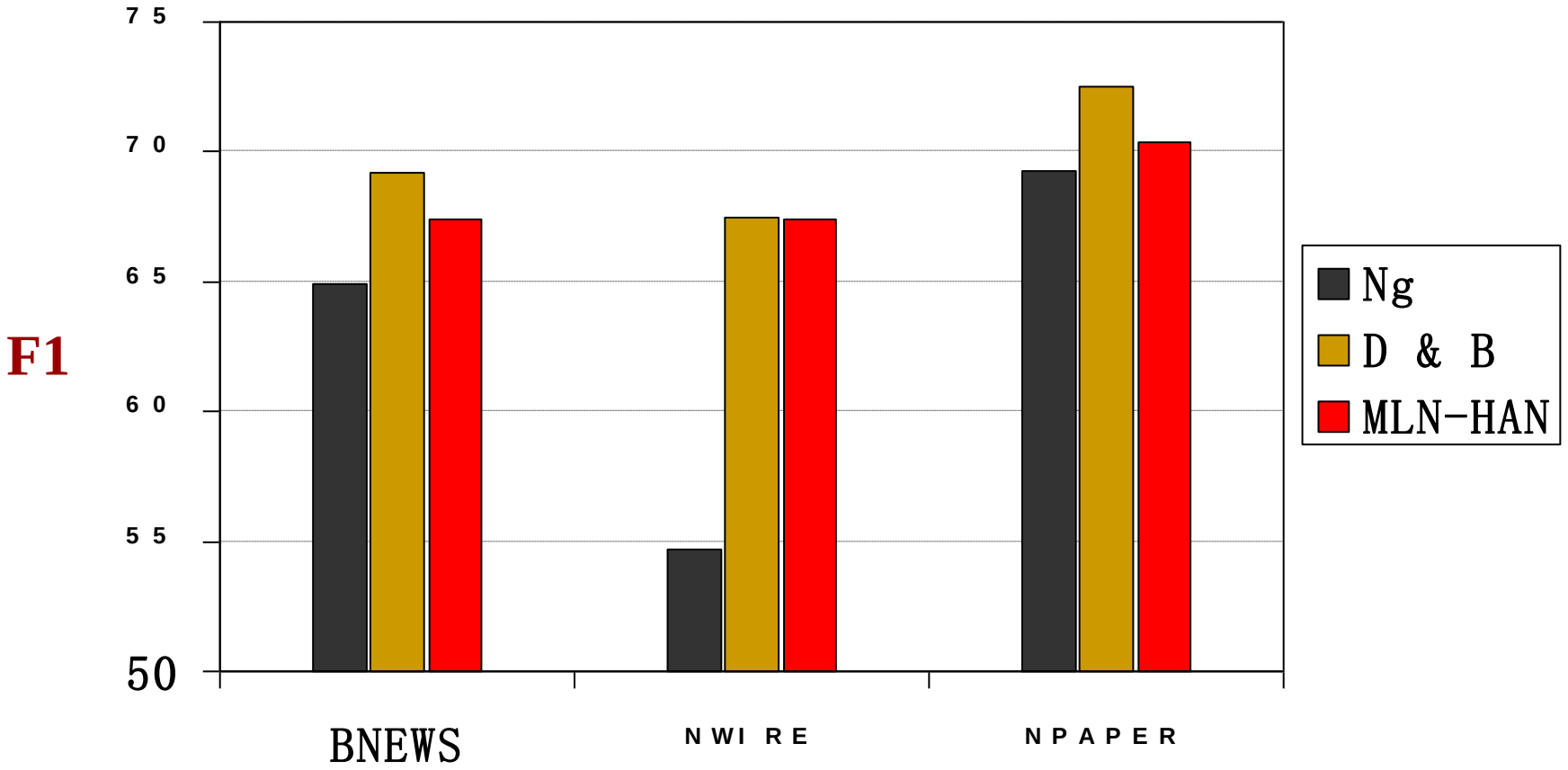
Results: MUC-6



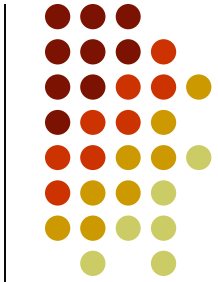
Results: ACE-2004



Results: ACE-2



Comparison with Previous Approaches



- Cluster-based
- Simpler modeling for salience
  $\Rightarrow$ Requires less training data
- Identify heads using head rules

  E.g., “the **President** of the **USA**”
- **Leverage joint inference**

  E.g., “Mr. **Bush**, the **President** …”

Error Analysis

- Features beyond the head

  E.g., the Finance Committee, the Defense Committee

- Speech pronouns, quotes, …

  E.g., I, we, you; “I am not Bush”, McCain said …

- Identify appositions and predicate nominals

  E.g., “Mike Sullivan, VOA News”

- Context and world knowledge

  E.g., “the White House”

Overview

- Motivation
- Background
- Markov logic
  - Inference
  - Learning
- Applications
  - Coreference resolution
- **Discussion**

Conclusion

- Pipeline architectures accumulate errors
- Joint inference is complex for human and machine
- Markov logic provides language and algorithms
  - Weighted first-order formulas → Markov network
  - Inference: Lifted belief propagation
  - Learning: Convex optimization and ILP
- Several successes to date
- First unsupervised coreference resolution system that rivals supervised ones
- Next steps: Combine more stages of the pipeline
- Open-source software: Alchemy

alchemy.cs.washington.edu