Online Large-Margin Training for SMT

Taro Watanabe, Jun Suzuki, Hajime Tsukada and Hideki Isozaki
NTT Communication Science Labs.

Overview

Overview

- MER Training Approach (Och, 2003):
  - Do not scale to large # of parameters.

Overview

- MER Training Approach (Och, 2003):
  - Do not scale to large # of parameters.
- Online Discriminative Training Approaches (Tillmann and Zhang, 2006; Liang et al., 2006):
  - Large # of parameters estimated on large data, but moderate improvements.

Overview

- MER Training Approach (Och, 2003):
  - Do not scale to large # of parameters.
- Online Discriminative Training Approaches (Tillmann and Zhang, 2006; Liang et al., 2006):
  - Large # of parameters estimated on large data, but moderate improvements.
- This work:
  - Online Large-Margin Training
  - Millions of parameters
  - Less than 1K sentences for training

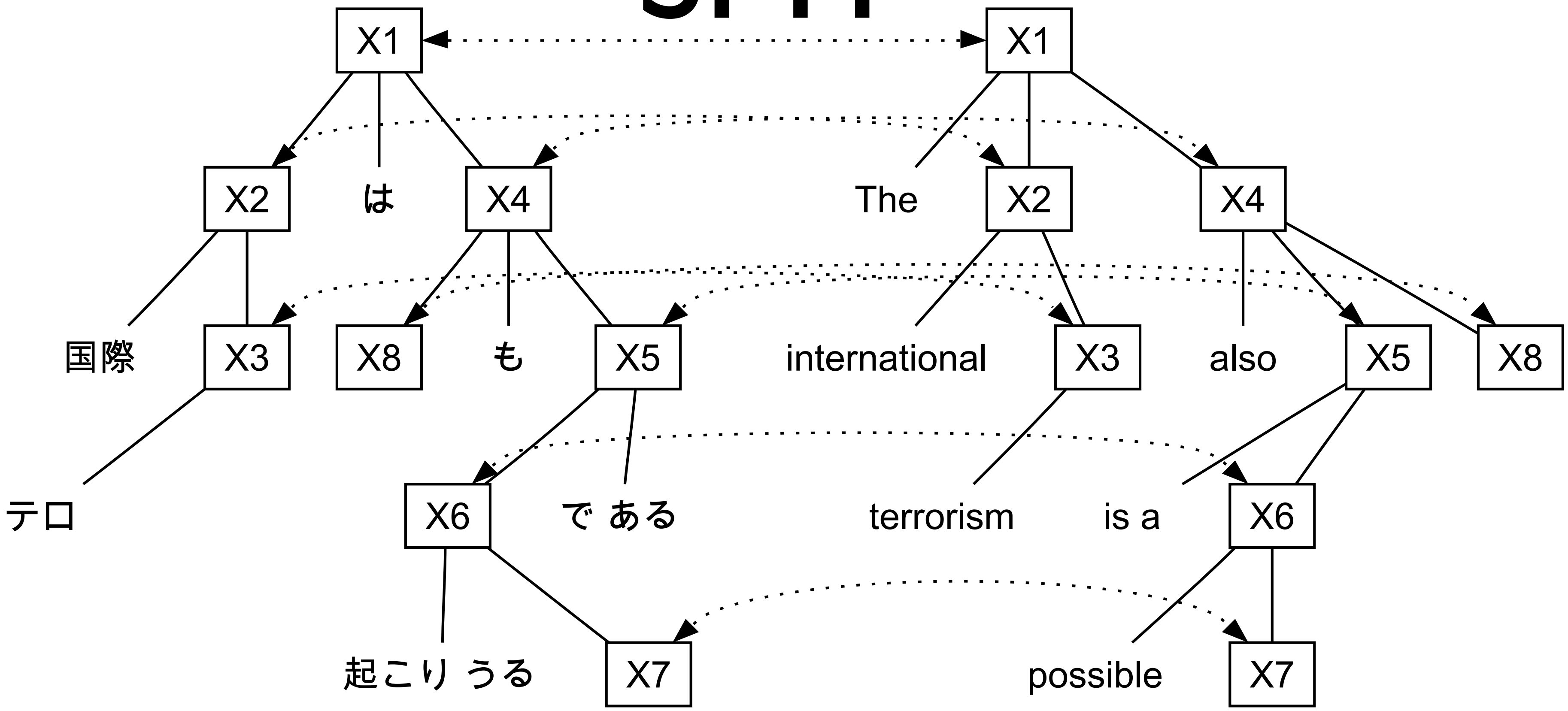
Statistical Machine Translation

$$\hat{e} = \operatorname*{argmax}_{e} \mathbf{w}^\top \cdot \mathbf{h}(f, e)$$

Hierarchical Phrase-based SMT

- Phrase embedded phrases via non-terminals (Chiang, 2005)
- An efficient top-down search (Watanabe et al., 2006)

Hierarchical Phrase-based SMT



- Phrase embedded phrases via non-terminals (Chiang, 2005)
- An efficient top-down search (Watanabe et al., 2006)

Features

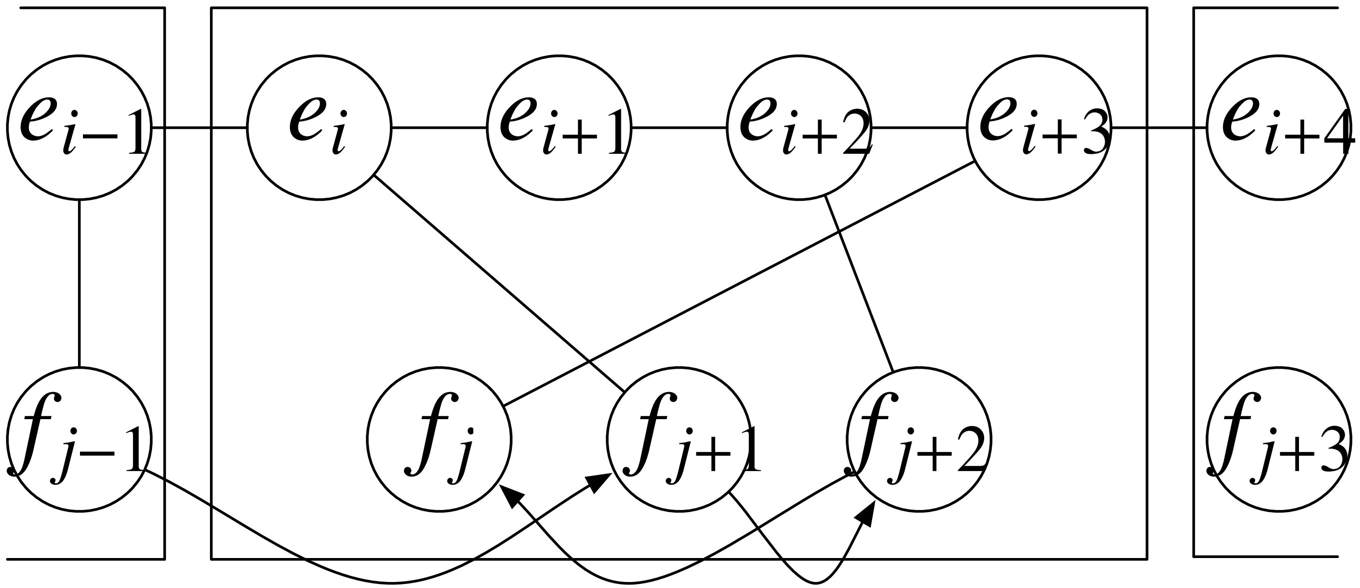
Features

- A standard Hiero-like features (Chiang, 2005):
  - n-gram language model, (hierarchical) phrase translation probabilities etc.
  - Phrase motivated penalties

Features

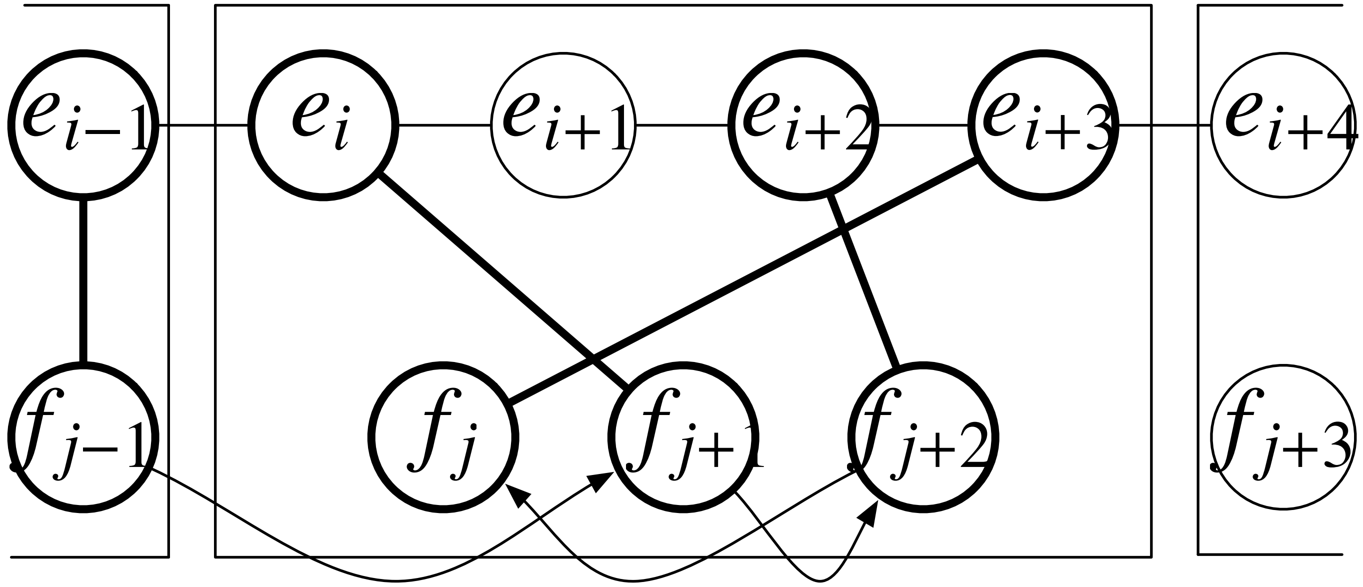
- A standard Hiero-like features (Chiang, 2005):
  - n-gram language model, (hierarchical) phrase translation probabilities etc.
  - Phrase motivated penalties
- Sparse features:
  - Unigram/bigram word pair features
  - Target bigram features
  - Insertion features
  - Hierarchical features

Sparse Features



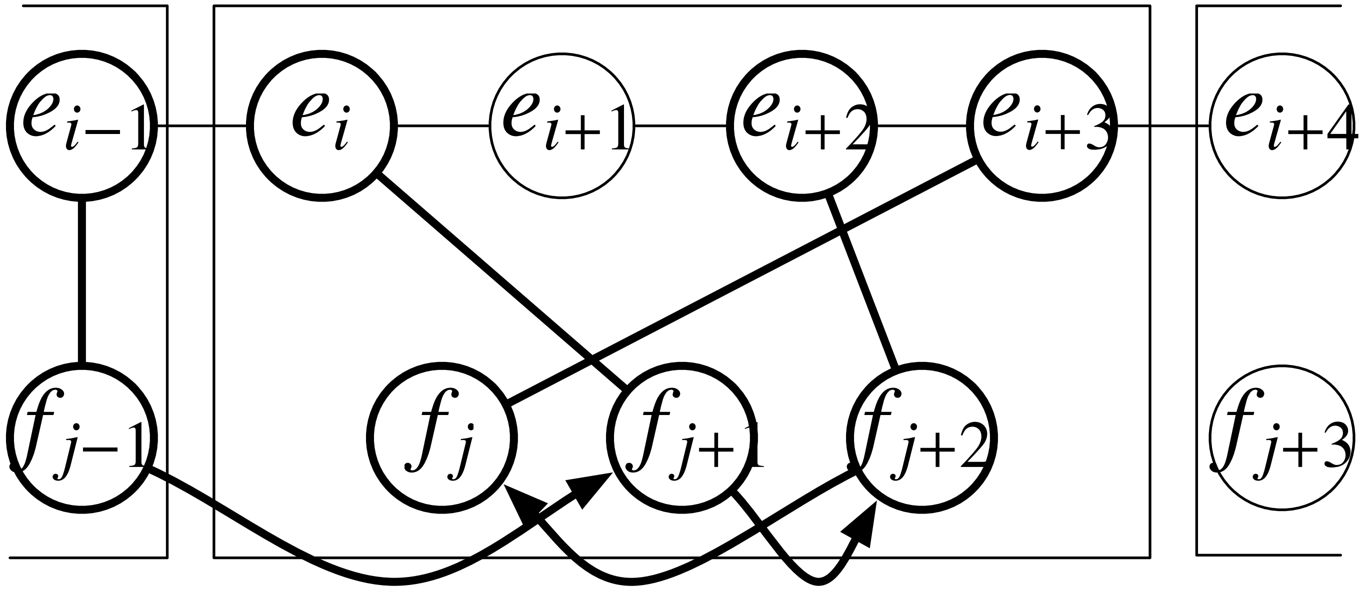
- Word pair features (unigram and bigram)
- Bigram features are ordered by the target side.

Sparse Features



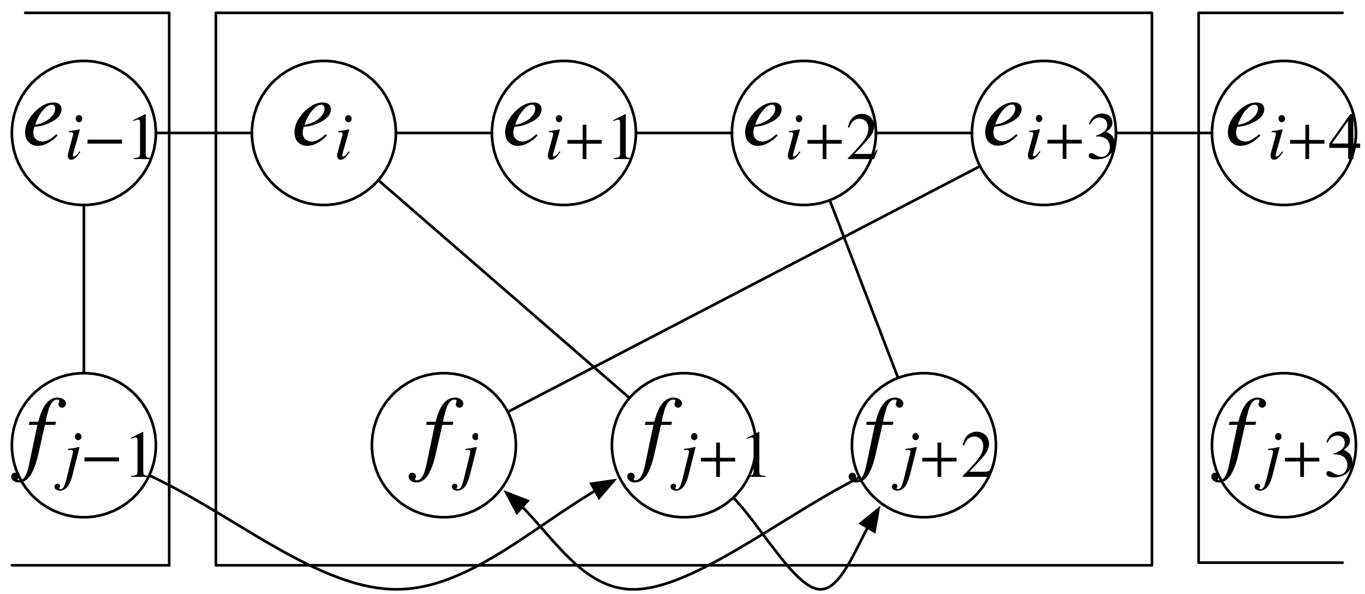
- Word pair features (unigram and bigram)
- Bigram features are ordered by the target side.

Sparse Features



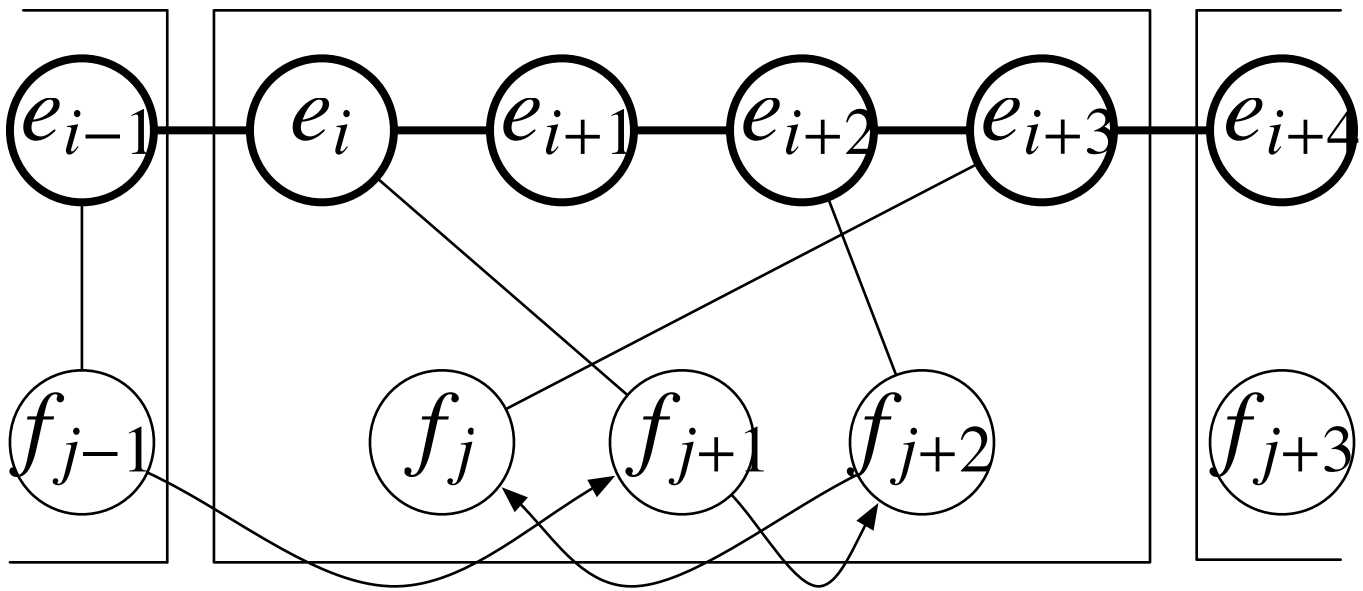
- Word pair features (unigram and bigram)
- Bigram features are ordered by the target side.

Sparse Features



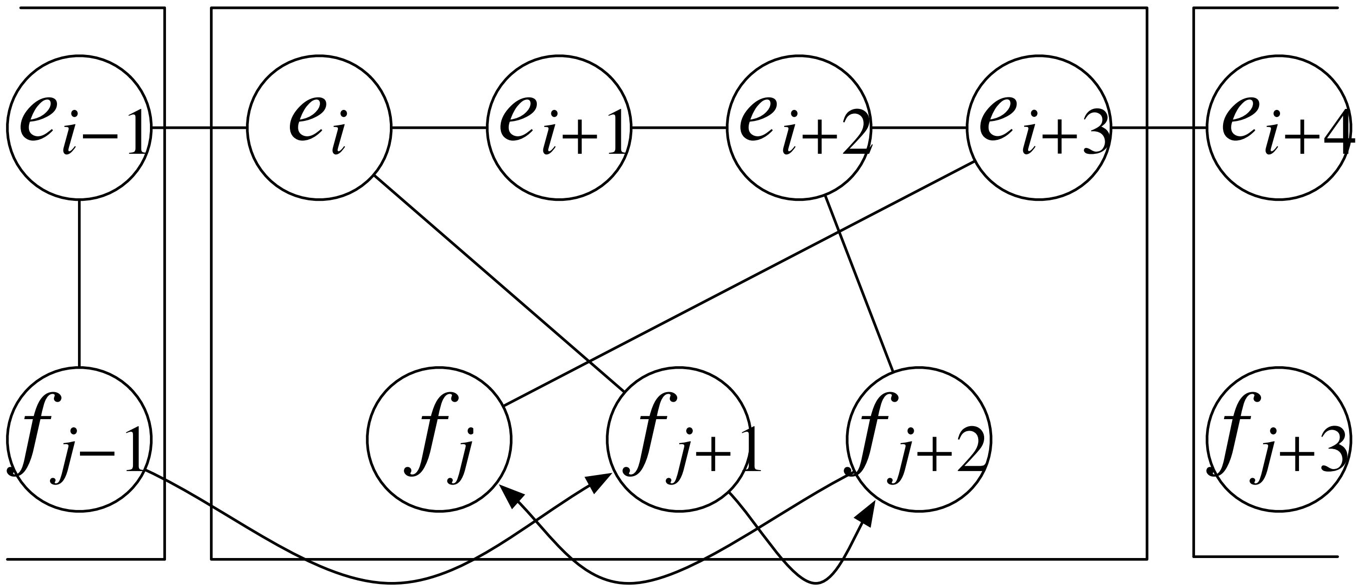
- Target bigram features.

Sparse Features



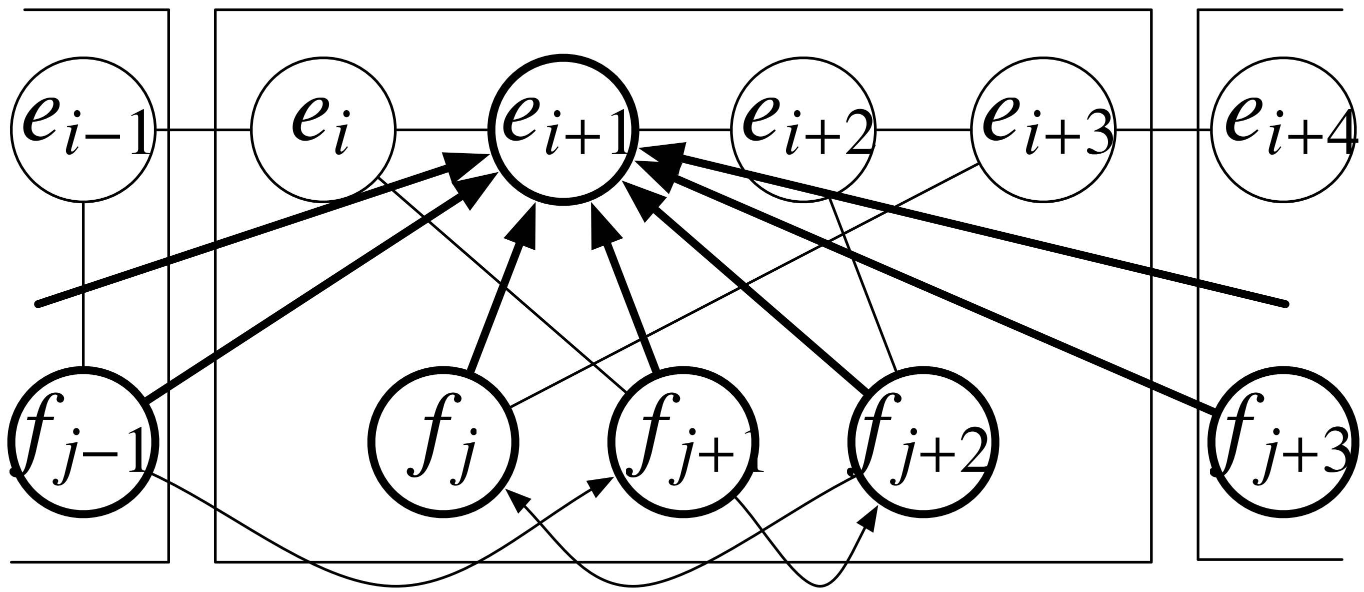
- Target bigram features.

Sparse Features



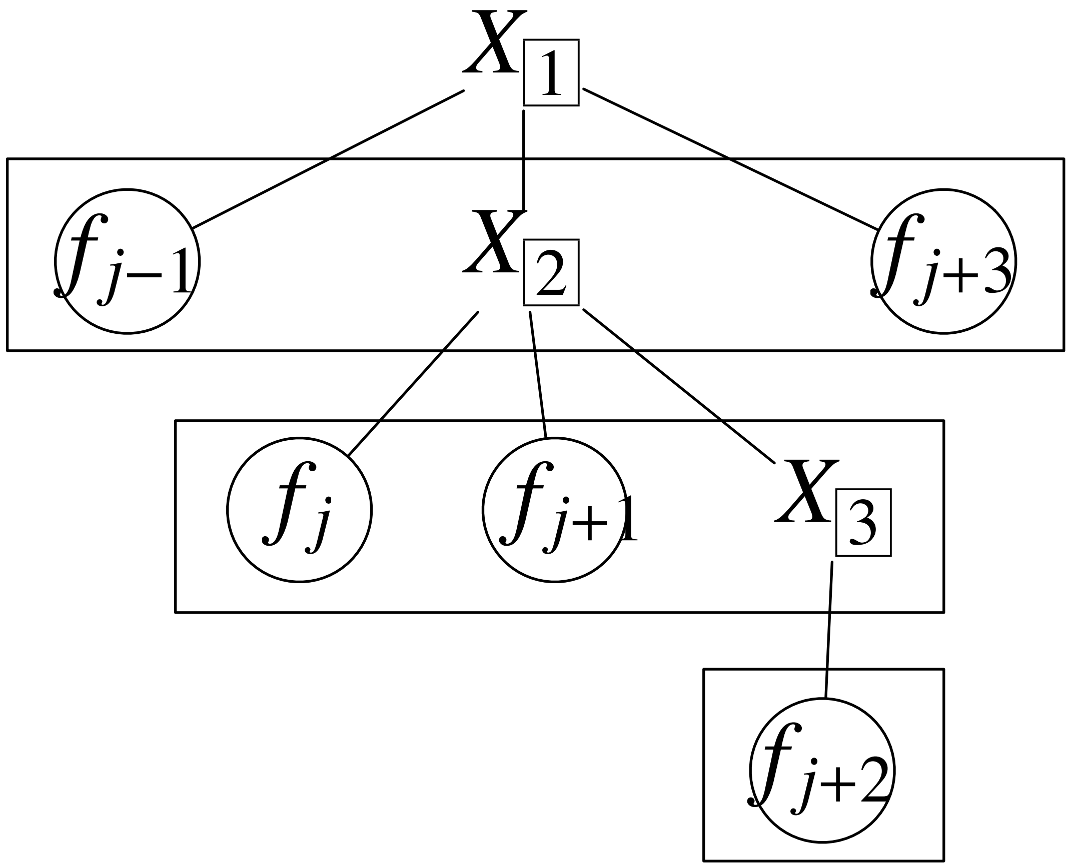
- Insertion features.
  - Each inserted word is associated with all the source words.

Sparse Features



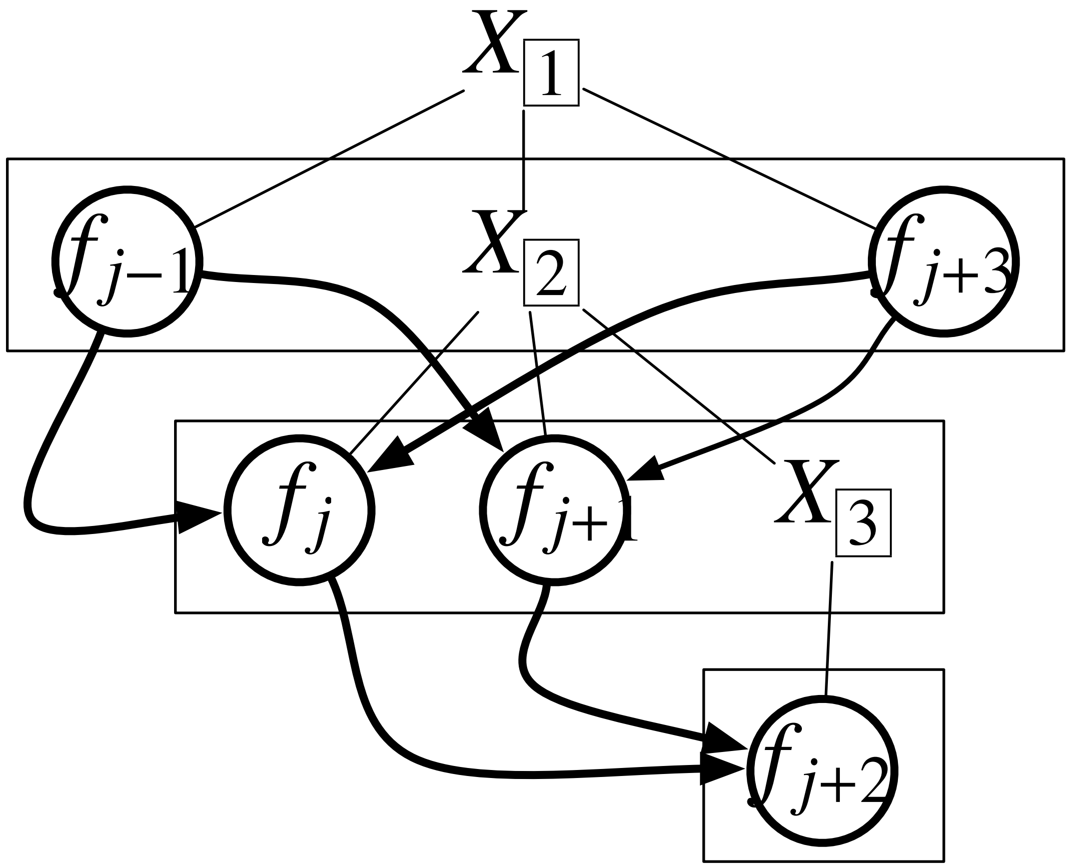
- Insertion features.
  - Each inserted word is associated with all the source words.

Sparse Features



- Hierarchical features.
  - Dependency structure on the source side.

Sparse Features



- Hierarchical features.
  - Dependency structure on the source side.

Sparse Features

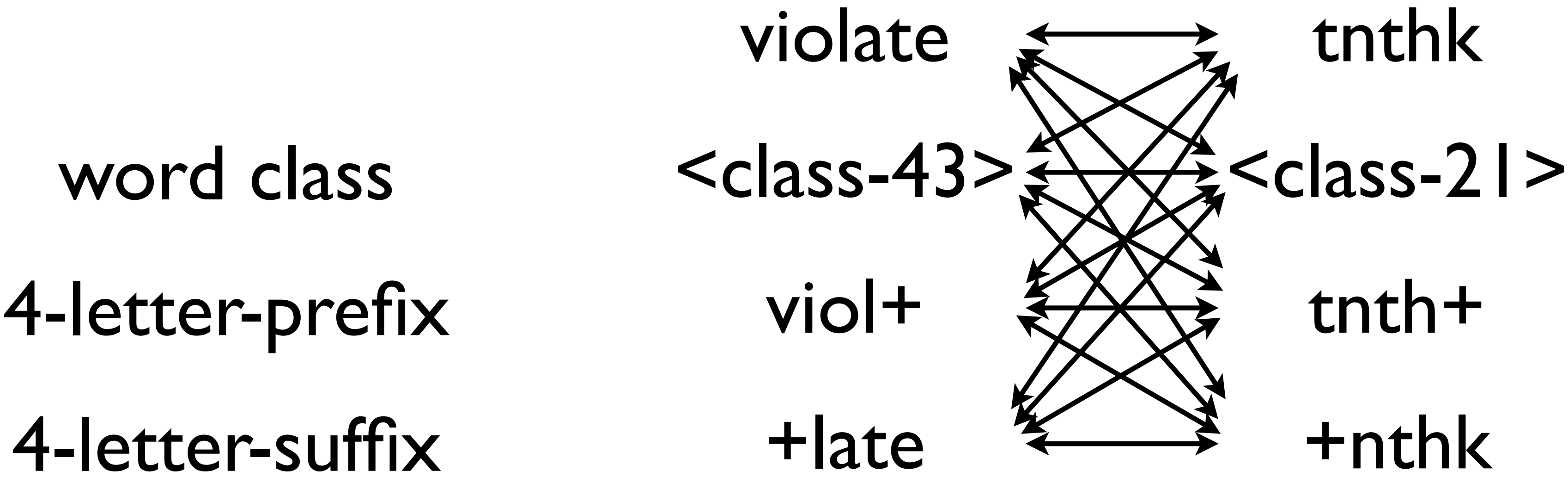
- Use of normalized tokens (POS/word class/prefix/etc.)
- Consider all possible combinations

Sparse Features

violate ⟷ tnthk

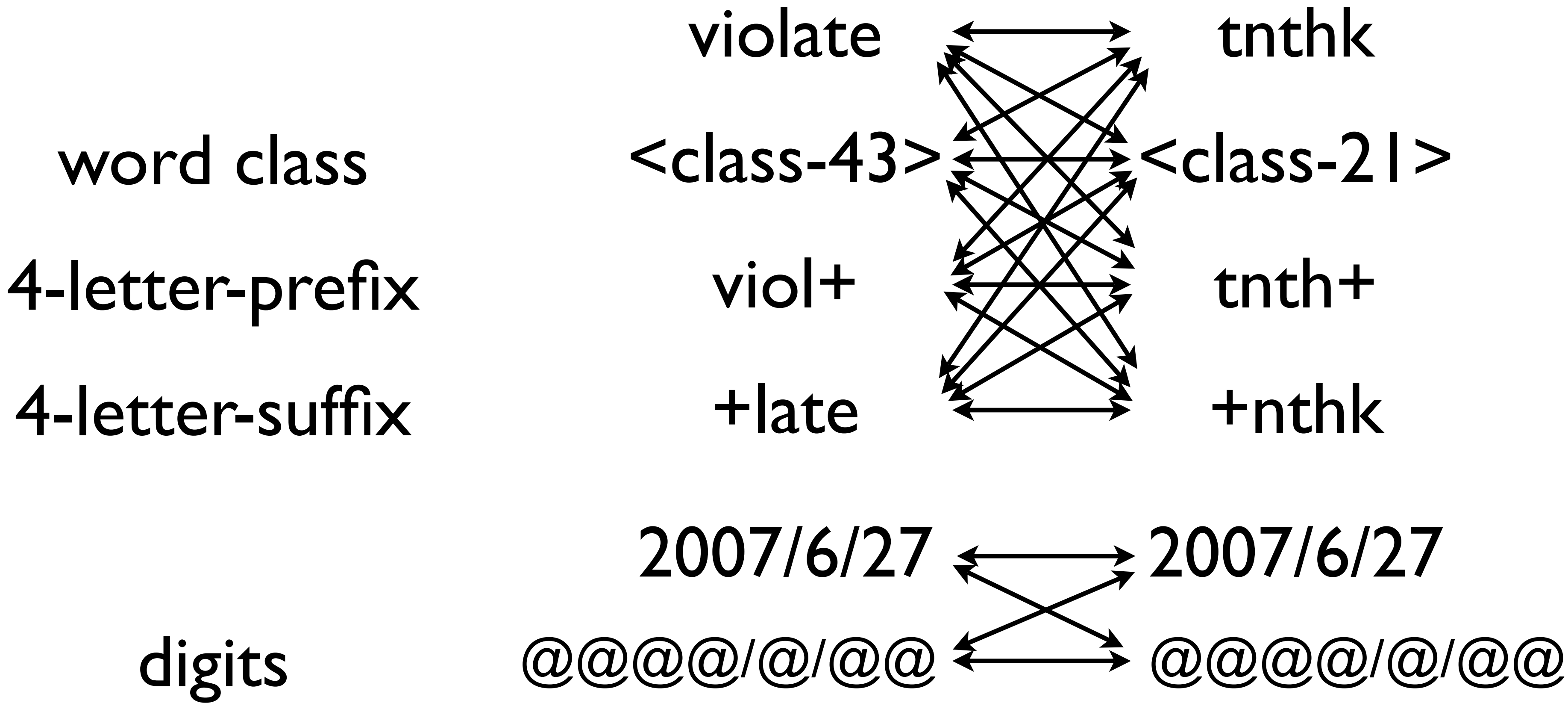
- Use of normalized tokens (POS/word class/prefix/etc.)
- Consider all possible combinations

Sparse Features



- Use of normalized tokens (POS/word class/prefix/etc.)
- Consider all possible combinations

Sparse Features



- Use of normalized tokens (POS/word class/prefix/etc.)
- Consider all possible combinations

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$

m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$

$i = 0$

for $n = 1, ..., N$ **do**

  for $t = 1, ..., T$ **do**

    $C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$

    $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$

    $\mathbf{w}^{i+1} = \text{update } \mathbf{w}^i \text{ using } C^t \text{ w.r.t. } \mathcal{O}^t$

    $i = i + 1$

  end for

end for

return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$
m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$
$i = 0$

for $n = 1, ..., N$ **do**
  for $t = 1, ..., T$ **do**
    $C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$
    $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$
    $\mathbf{w}^{i+1} = \text{update } \mathbf{w}^i \text{ using } C^t \text{ w.r.t. } \mathcal{O}^t$
    $i = i + 1$
  end for
end for
return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$
m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$
$i = 0$

for $n = 1, ..., N$ **do**
  for $t = 1, ..., T$ **do**
    $C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$
    $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$
    $\mathbf{w}^{i+1}$ = update $\mathbf{w}^i$ using C^t w.r.t. $\mathcal{O}^t$
    $i = i + 1$
  end for
end for
return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$

Tillmann and Zhang (2006) precomputed oracles.

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$

m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$

$i = 0$

for $n = 1, ..., N$ **do**

  for $t = 1, ..., T$ **do**

    $C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$

    $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$

    $\mathbf{w}^{i+1} = \text{update } \mathbf{w}^i \text{ using } C^t \text{ w.r.t. } \mathcal{O}^t$

    $i = i + 1$

  end for

end for

return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$

m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$

$i = 0$

for $n = 1, ..., N$ **do**

  for $t = 1, ..., T$ **do**

    $C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$

    $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$

    $\mathbf{w}^{i+1}$ = update $\mathbf{w}^i$ using C^t w.r.t. $\mathcal{O}^t$

    $i = i + 1$

  end for

end for

return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$
m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$
$i = 0$

```
for n = 1, ..., N do
    for t = 1, ..., T do
        C^t ← best_k(f^t; w^i)
        O^t ← oracle_m(O^t ∪ C^t; e^t)
        w^{i+1} = update w^i using C^t w.r.t. O^t
        i = i + 1
    end for
end for
return (Σ_{i=1}^{NT} w^i) / NT
```

$$\text{for } n = 1, \ldots, N \text{ do}$$
$$\quad \text{for } t = 1, \ldots, T \text{ do}$$
$$\quad\quad C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$$
$$\quad\quad \mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$$
$$\quad\quad \mathbf{w}^{i+1} = \text{update } \mathbf{w}^i \text{ using } C^t \text{ w.r.t. } \mathcal{O}^t$$
$$\quad\quad i = i + 1$$
$$\quad \text{end for}$$
$$\text{end for}$$
$$\text{return } \frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$$

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$

m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$

$i = 0$

for $n = 1, ..., N$ **do**

  for $t = 1, ..., T$ **do**

    $C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$

    $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$

    $\mathbf{w}^{i+1} = \text{update } \mathbf{w}^i \text{ using } C^t \text{ w.r.t. } \mathcal{O}^t$

    $i = i + 1$

  end for

end for

return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$

Liang et al. (2006) discarded possibly better oracles.

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$

m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$

$i = 0$

for $n = 1, ..., N$ **do**

  for $t = 1, ..., T$ **do**

    $\mathcal{C}^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$

    $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup \mathcal{C}^t; \mathbf{e}^t)$

    $\mathbf{w}^{i+1} = \text{update } \mathbf{w}^i \text{ using } \mathcal{C}^t \text{ w.r.t. } \mathcal{O}^t$

    $i = i + 1$

  end for

end for

return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$

Online Training Algorithm

Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$

m-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$

$i = 0$

for $n = 1, ..., N$ **do**

  for $t = 1, ..., T$ **do**

    $C^t \leftarrow \text{best}_k(f^t; \mathbf{w}^i)$

    $\mathcal{O}^t \leftarrow \text{oracle}_m(\mathcal{O}^t \cup C^t; \mathbf{e}^t)$

    $\mathbf{w}^{i+1}$ = update $\mathbf{w}^i$ using C^t w.r.t. $\mathcal{O}^t$

    $i = i + 1$

  end for

end for

return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$

Online Large-Margin Training

$$\hat{\mathbf{w}}^{i+1} = \operatorname*{argmin}_{\mathbf{w}^{i+1}} \frac{1}{2} \|\mathbf{w}^{i+1} - \mathbf{w}^{i}\|^2$$

subject to

$$s^{i+1}(f^t, \hat{e}) - s^{i+1}(f^t, e') \geq L(\hat{e}, e'; \mathbf{e}^t)$$

$$\xi(\hat{e}, e') \geq 0$$

$$\forall \hat{e} \in O^t, \forall e' \in C^t$$

Online Large-Margin Training

$$\hat{\mathbf{w}}^{i+1} = \underset{\mathbf{w}^{i+1}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}^{i+1} - \mathbf{w}^{i}\|^2 + C \sum_{\hat{e},e'} \xi(\hat{e}, e')$$

subject to

$$s^{i+1}(f^t, \hat{e}) - s^{i+1}(f^t, e') + \xi(\hat{e}, e') \geq L(\hat{e}, e'; \mathbf{e}^t)$$

$$\xi(\hat{e}, e') \geq 0$$

$$\forall \hat{e} \in O^t, \forall e' \in C^t$$

Online Large-Margin Training

$$\hat{\mathbf{w}}^{i+1} = \operatorname*{argmin}_{\mathbf{w}^{i+1}} \frac{1}{2}\|\mathbf{w}^{i+1} - \mathbf{w}^{i}\|^2 + C \sum_{\hat{e},e'} \xi(\hat{e}, e')$$

subject to

$$s^{i+1}(f^t, \hat{e}) - s^{i+1}(f^t, e') + \xi(\hat{e}, e') \geq L(\hat{e}, e'; \mathbf{e}^t)$$

$$\xi(\hat{e}, e') \geq 0$$

$$\forall \hat{e} \in O^t, \forall e' \in C^t$$

- Constrained by m-oracle + k-best.
- "C" to control the amount of updates.

Parameter Updates

$$\mathbf{w}^{i+1} = \mathbf{w}^{i} + \sum_{\hat{e},e'} \alpha(\hat{e}, e') \Big(\mathbf{h}(f^{t}, \hat{e}) - \mathbf{h}(f^{t}, e')\Big)$$

Parameter Updates

$$\mathbf{w}^{i+1} = \mathbf{w}^i + \sum_{\hat{e},e'} \alpha(\hat{e}, e') \left(\mathbf{h}(f^t, \hat{e}) - \mathbf{h}(f^t, e')\right)$$

- This work (1-oracle + 1-best)

$$\alpha = \max\left(0, \min\left(C, \frac{L(\hat{e}, e'; \mathbf{e}^t) - \left(s^i(f^t, \hat{e}) - s^i(f^t, e')\right)}{\|\mathbf{h}(f^t, \hat{e}) - \mathbf{h}(f^t, e')\|^2}\right)\right)$$

Parameter Updates

$$\mathbf{w}^{i+1} = \mathbf{w}^{i} + \sum_{\hat{e},e'} \alpha(\hat{e}, e') \Big(\mathbf{h}(f^t, \hat{e}) - \mathbf{h}(f^t, e')\Big)$$

- This work (1-oracle + 1-best)

$$\alpha = \max\left(0, \min\left(C, \frac{L(\hat{e}, e'; \mathbf{e}^t) - \Big(s^i(f^t, \hat{e}) - s^i(f^t, e')\Big)}{\|\mathbf{h}(f^t, \hat{e}) - \mathbf{h}(f^t, e')\|^2}\right)\right)$$

- Perceptron Training (Liang et al., 2006)

$$\alpha = 1$$

Parameter Updates

$$\mathbf{w}^{i+1} = \mathbf{w}^{i} + \sum_{\hat{e},e'} \alpha(\hat{e}, e') \Big(\mathbf{h}(f^t, \hat{e}) - \mathbf{h}(f^t, e') \Big)$$

- This work (1-oracle + 1-best)

$$\alpha = \max\left(0, \min\left(C, \frac{L(\hat{e}, e'; \mathbf{e}^t) - \Big(s^i(f^t, \hat{e}) - s^i(f^t, e') \Big)}{\|\mathbf{h}(f^t, \hat{e}) - \mathbf{h}(f^t, e')\|^2}\right)\right)$$

- Perceptron Training (Liang et al., 2006)

$$\alpha = 1$$

- SGD Training (Tillmann and Zhand, 2006)

$$\alpha = \eta L(\hat{e}, e'; \mathbf{e}^t) \cdot \max\Big(0, 1 - \Big(s^i(f^t, \hat{e}) - s^i(f^t, e') \Big)\Big)$$

Approximated BLEU

Approximated BLEU

- Document-BLEU or sentence-BLEU?

$$\mathrm{BLEU}(E; \mathbf{E}) = \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n(E, \mathbf{E})\right) \cdot \mathrm{BP}(E, \mathbf{E})$$

Approximated BLEU

- Document-BLEU or sentence-BLEU?

$$\mathrm{BLEU}(E;\mathbf{E}) = \exp\left(\frac{1}{N}\sum_{n=1}^{N}\log p_n(E,\mathbf{E})\right)\cdot \mathrm{BP}(E,\mathbf{E})$$

- Our method: compute the difference from an oracle BLEU (Watanabe et al., 2006)

$$\mathrm{BLEU}(\{\hat{e}^1, ..., \hat{e}^{t-1}, {\color{blue}e'}, \hat{e}^{t+1}, ..., \hat{e}^T\};\mathbf{E})$$

- Loss by an approximated BLEU ≈ doument-wise loss.

Evaluation

Evaluation

- A standard NIST Arabic/English Translation
  - Hierarchical phrases from 3.8M sentences
  - 5-gram from English Gigaword
  - Trained on MT 2003, tested on MT 2004/2005

Evaluation

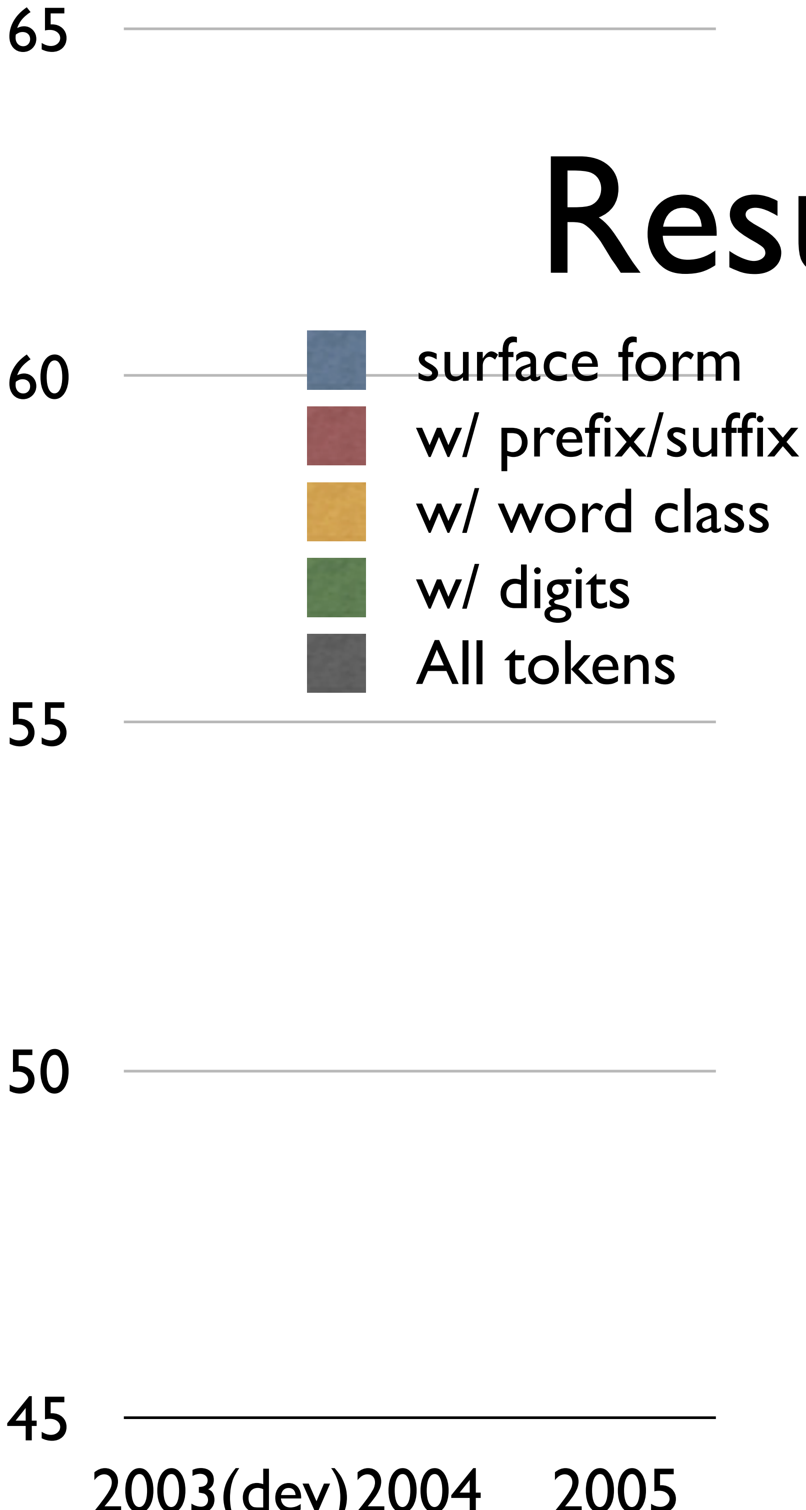
- A standard NIST Arabic/English Translation
  - Hierarchical phrases from 3.8M sentences
  - 5-gram from English Gigaword
  - Trained on MT 2003, tested on MT 2004/2005
- Experiments on (10-oracle, 10-best, 50 iterations):
  - Token-types
  - Structural features
  - m-oracle + k-best constraints

Evaluation

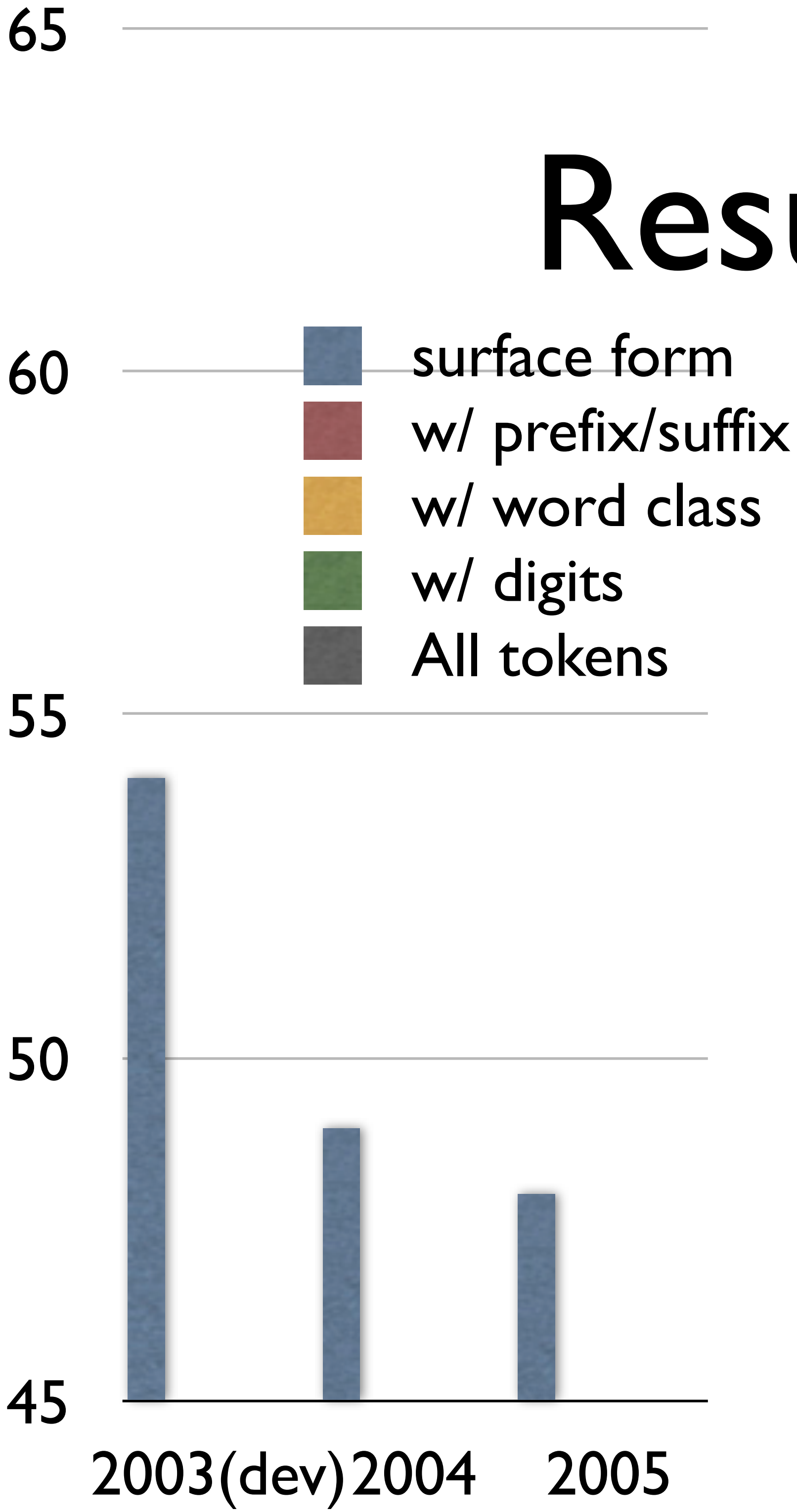
- A standard NIST Arabic/English Translation
  - Hierarchical phrases from 3.8M sentences
  - 5-gram from English Gigaword
  - Trained on MT 2003, tested on MT 2004/2005
- Experiments on (10-oracle, 10-best, 50 iterations):
  - Token-types
  - Structural features
  - m-oracle + k-best constraints
- 0.5M to 14M active features

Results (BLEU)

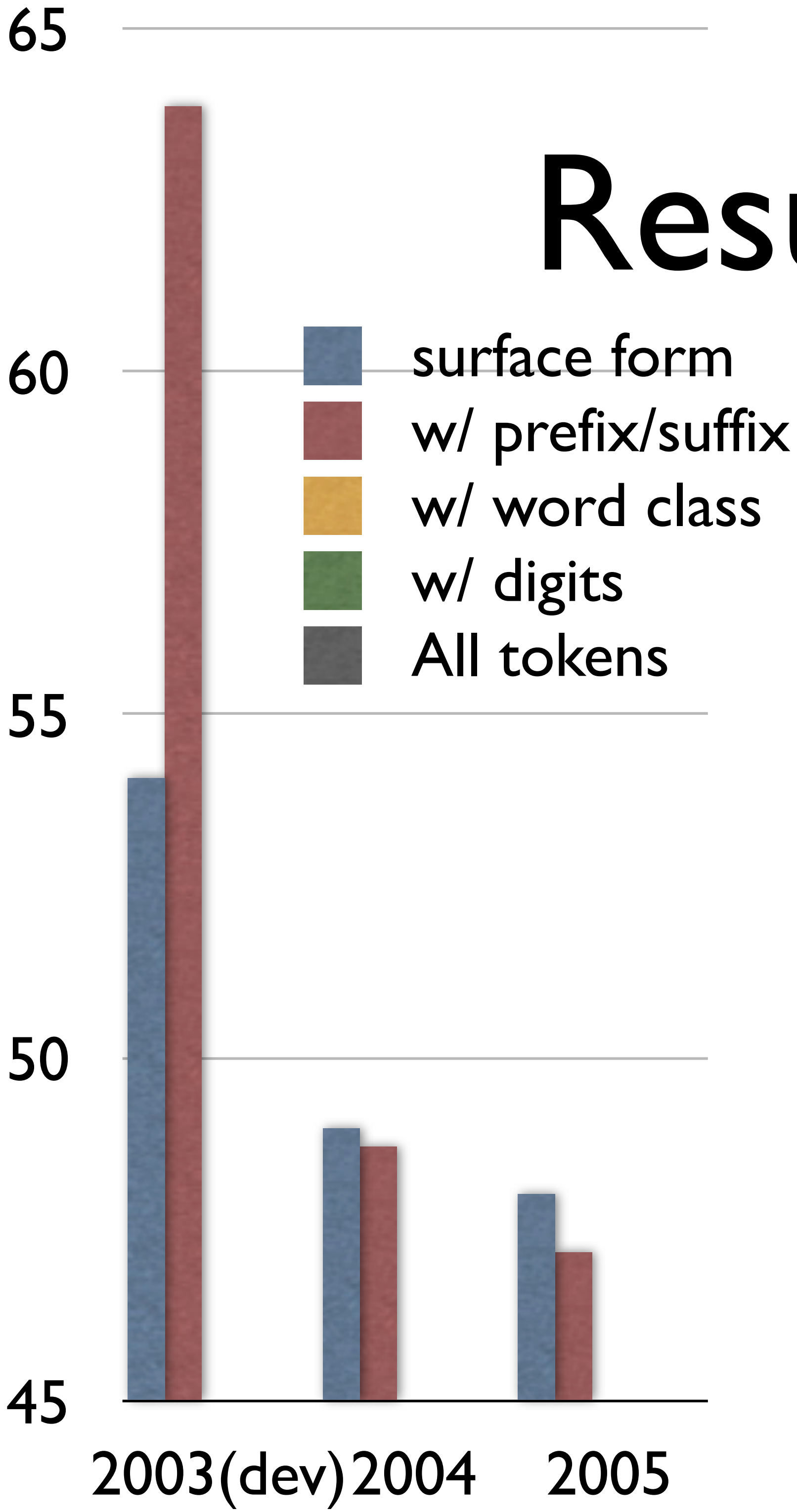
Results (BLEU)



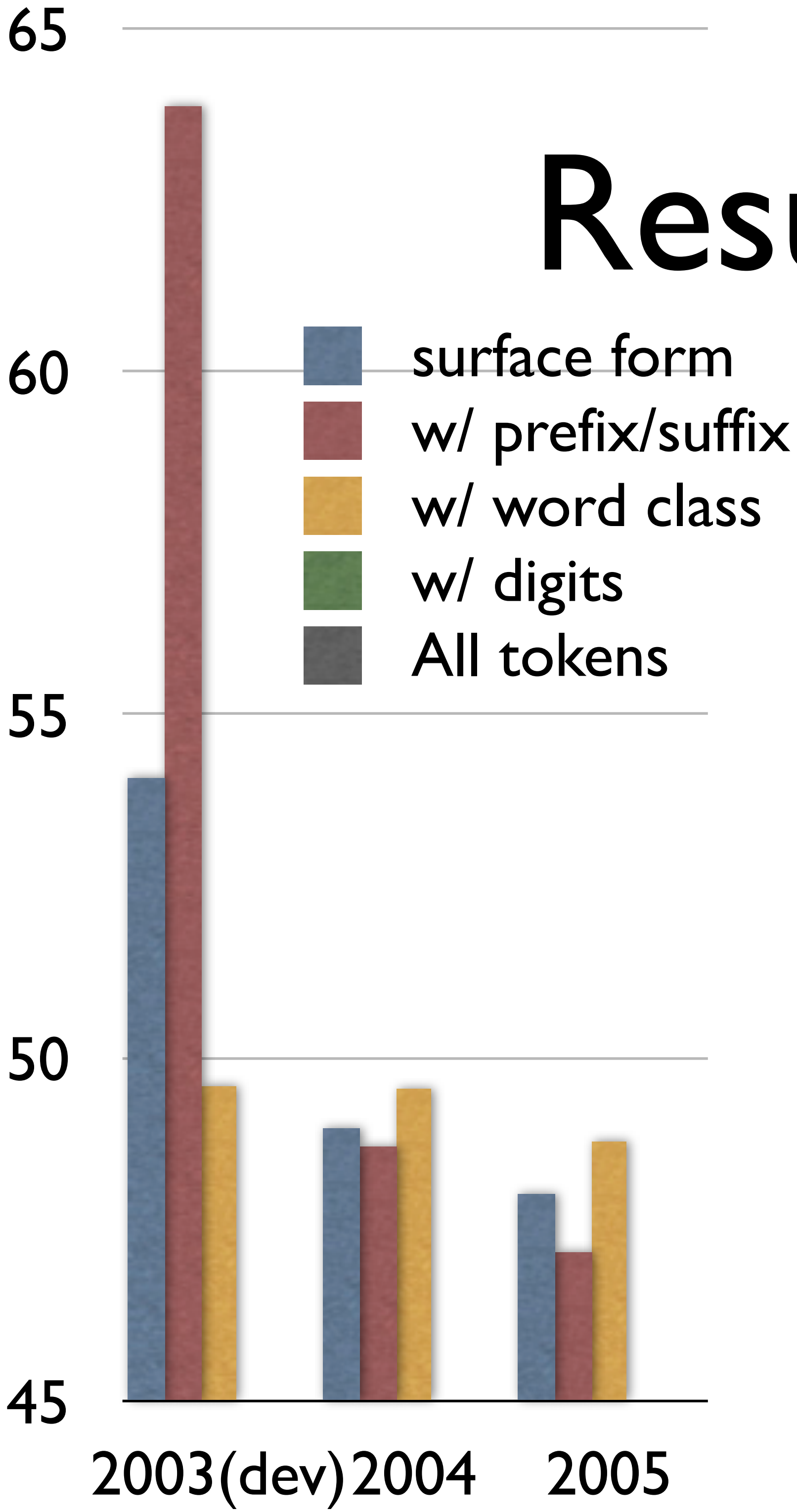
Results (BLEU)



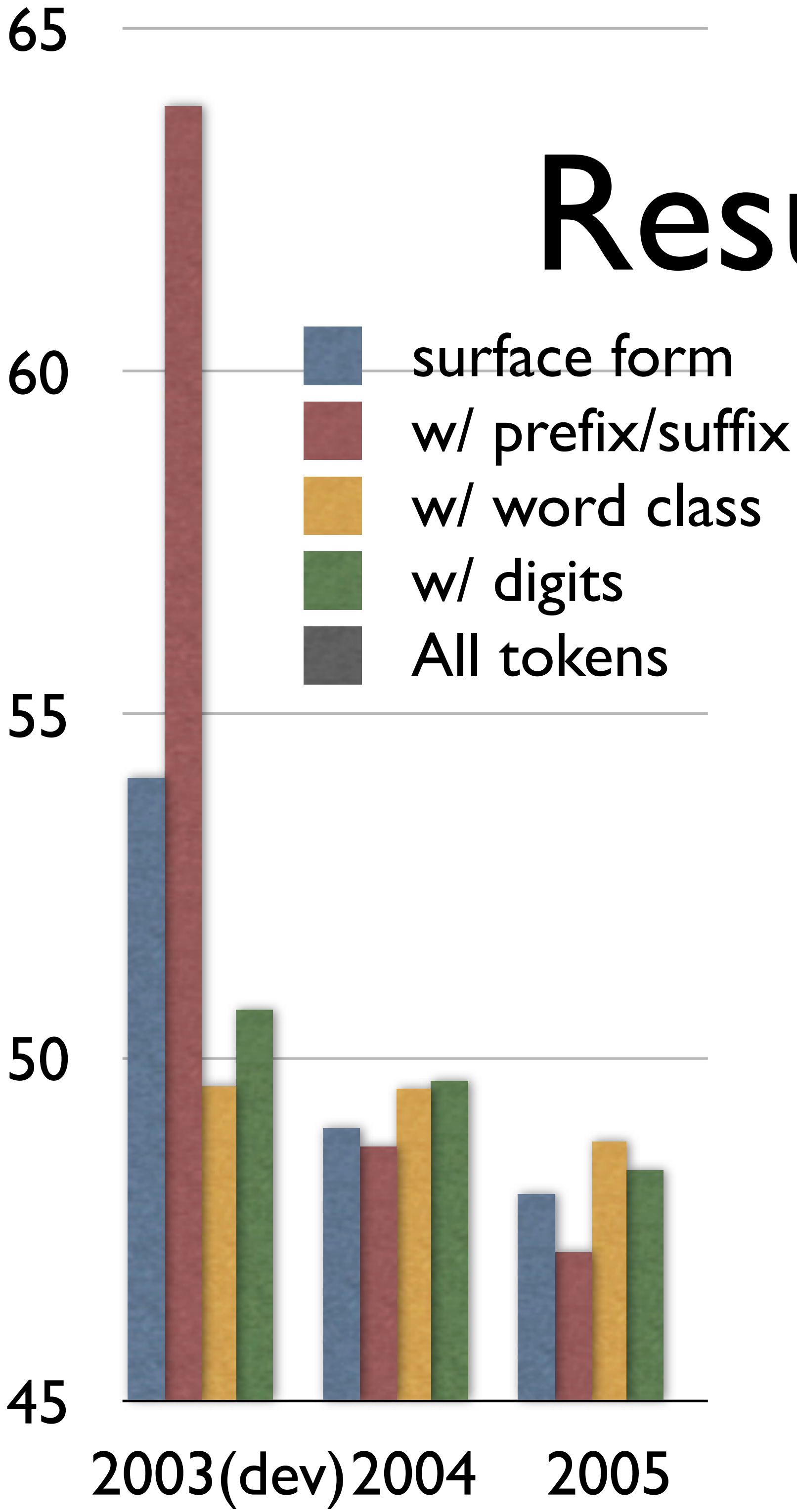
Results (BLEU)



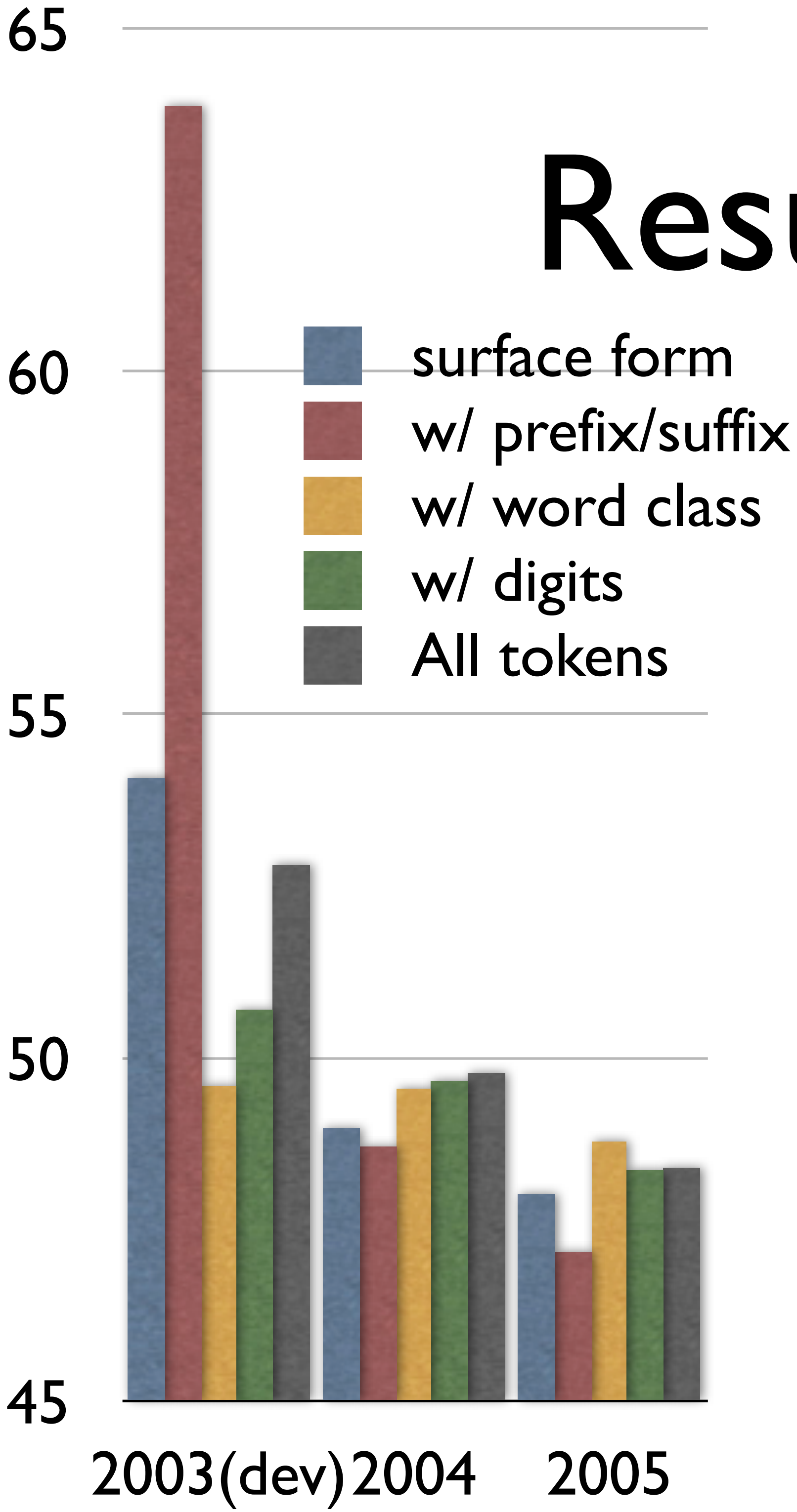
Results (BLEU)



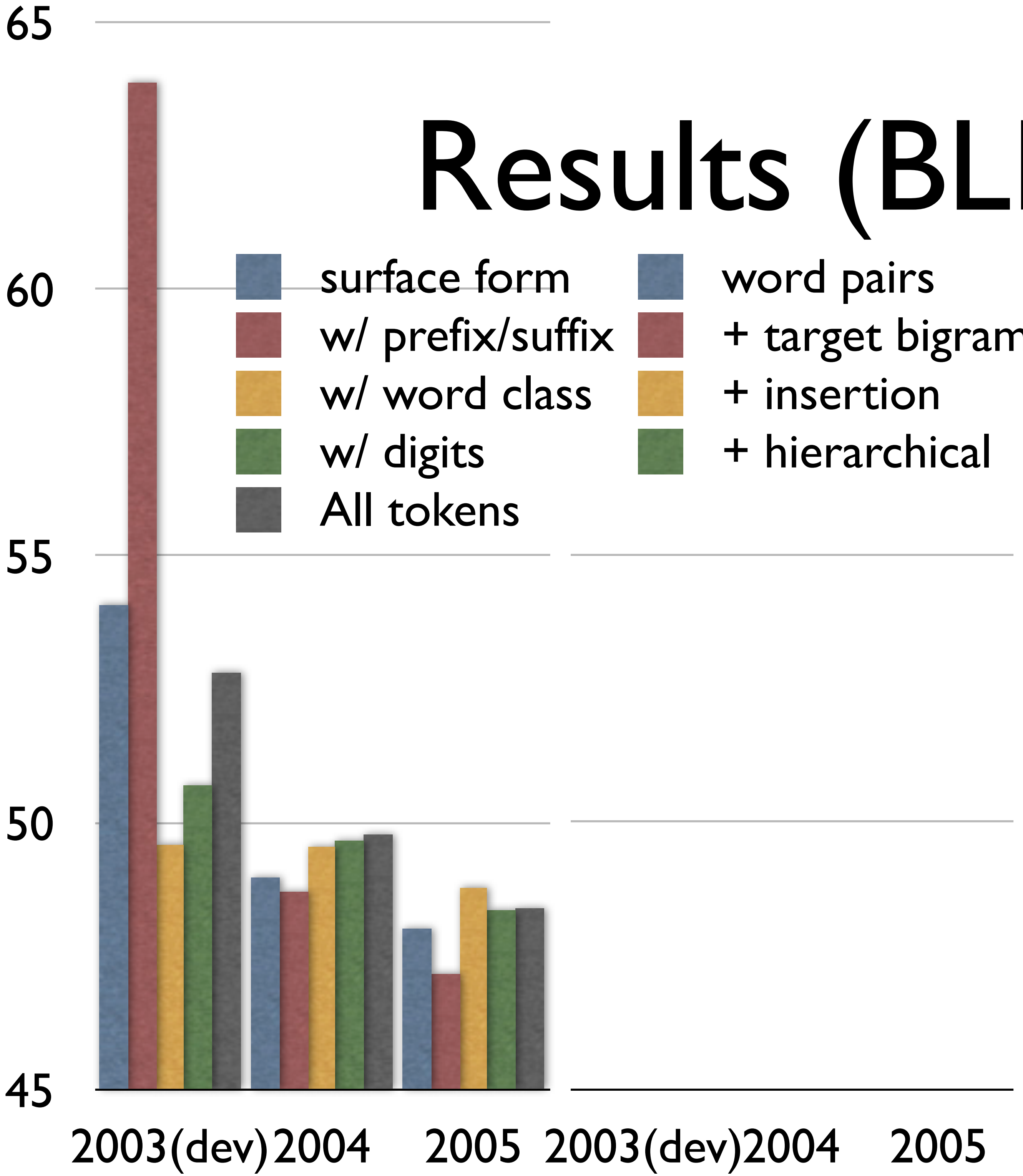
Results (BLEU)

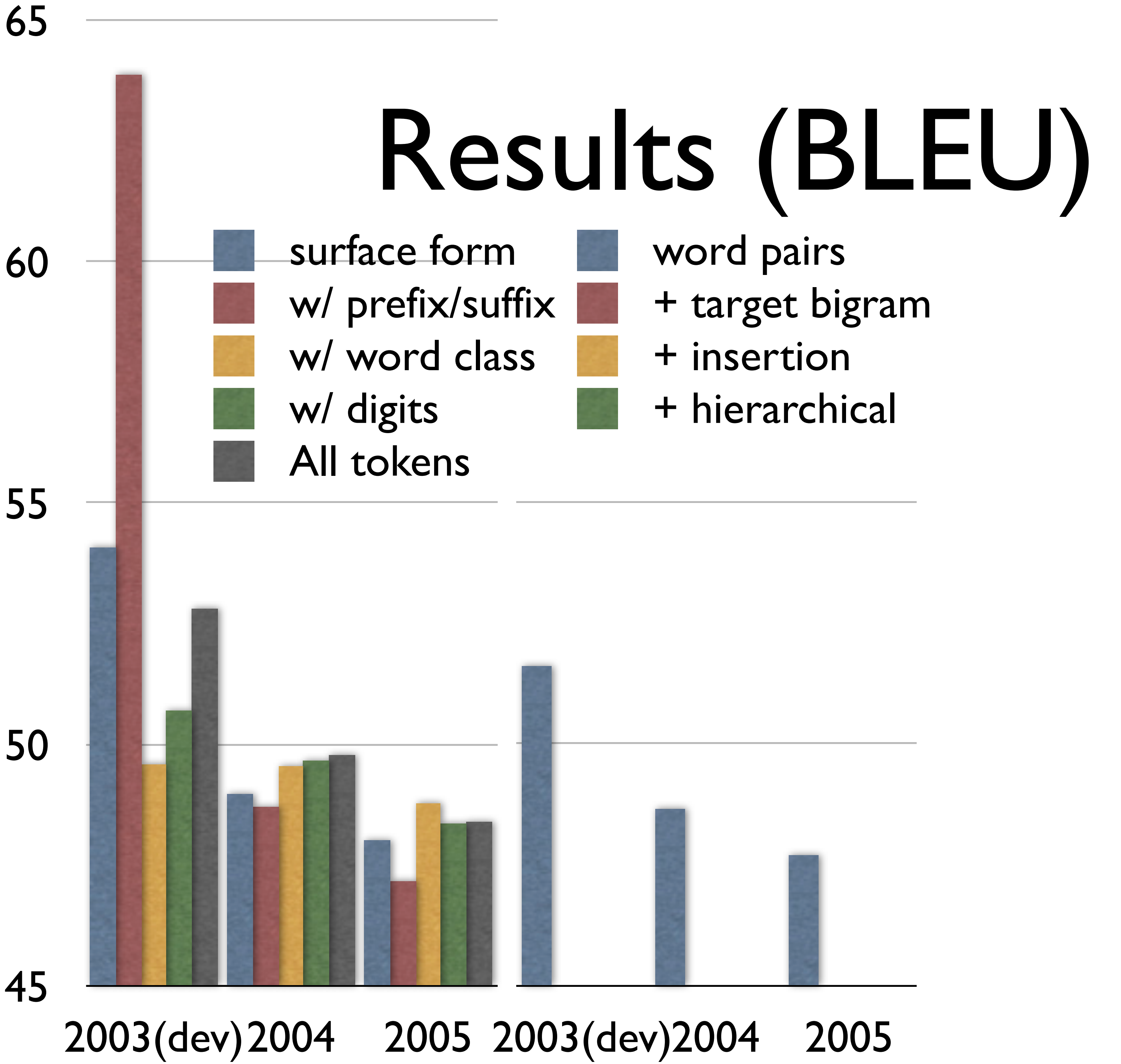


Results (BLEU)

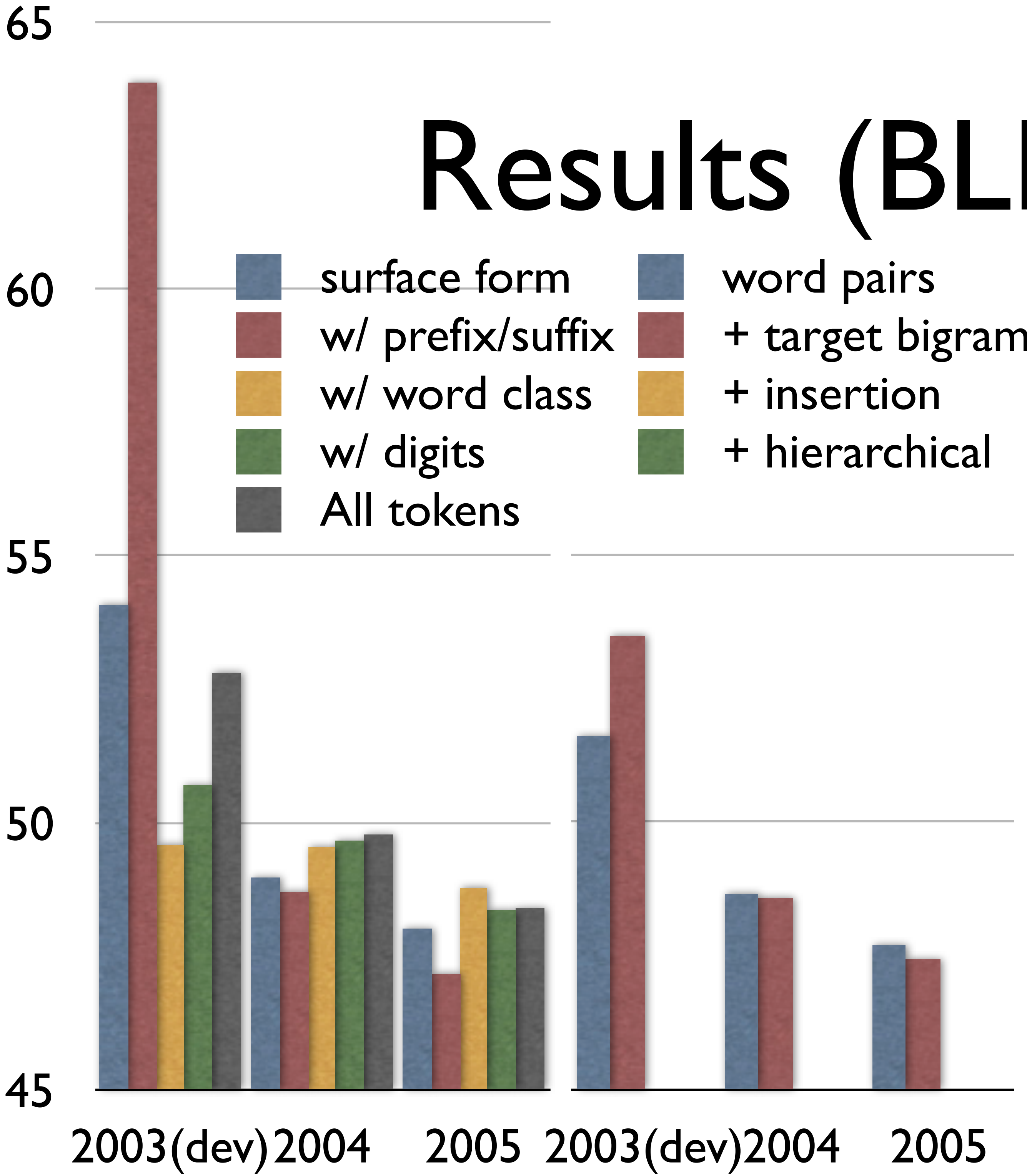


Results (BLEU)

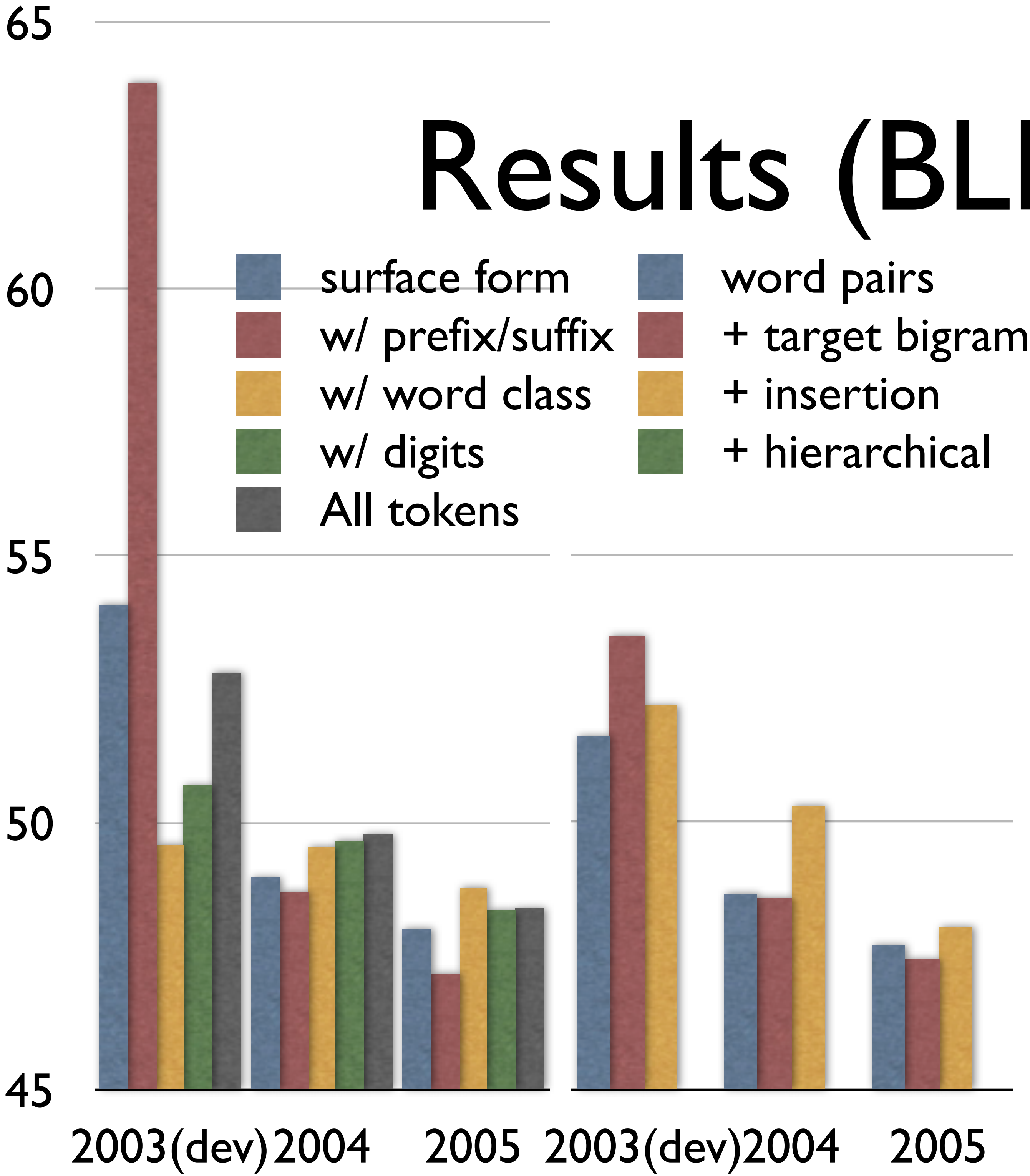



Results (BLEU)
surface form
w/ prefix/suffix
w/ word class
w/ digits
All tokens
word pairs
+ target bigram
+ insertion
+ hierarchical
65
60
55
50
45
2003(dev)
2004
2005
2003(dev)
2004
2005

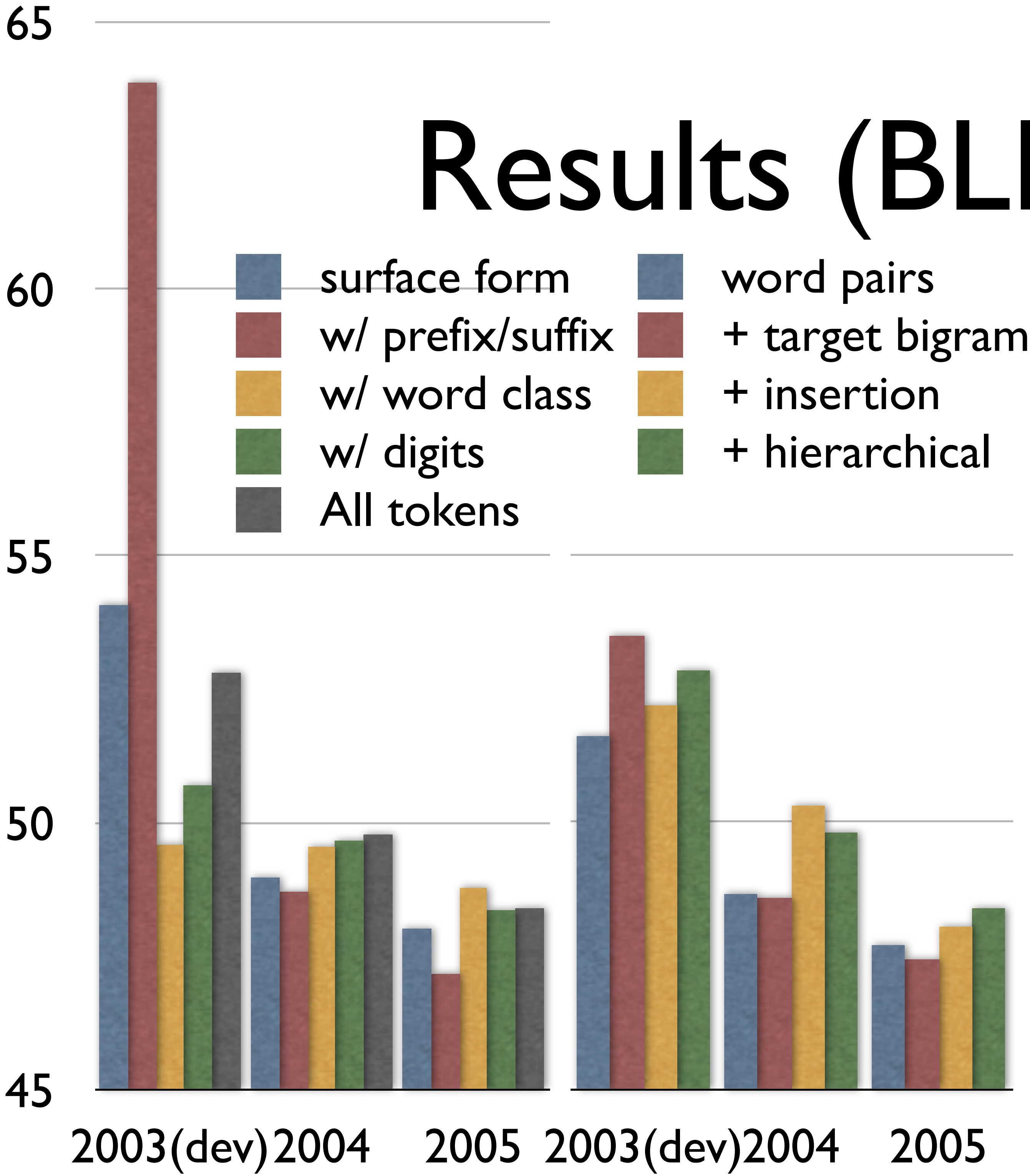
Results (BLEU)



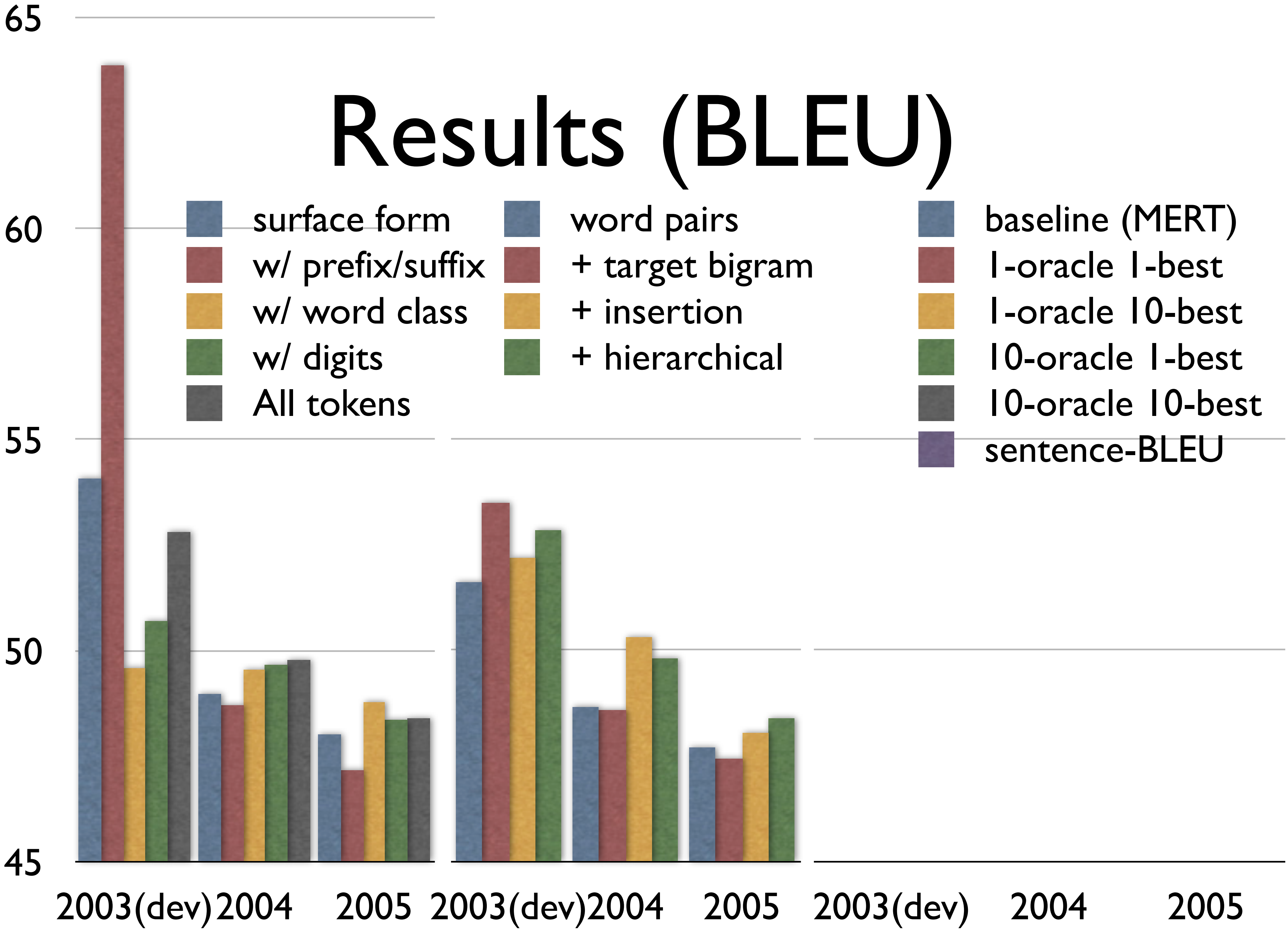
Results (BLEU)



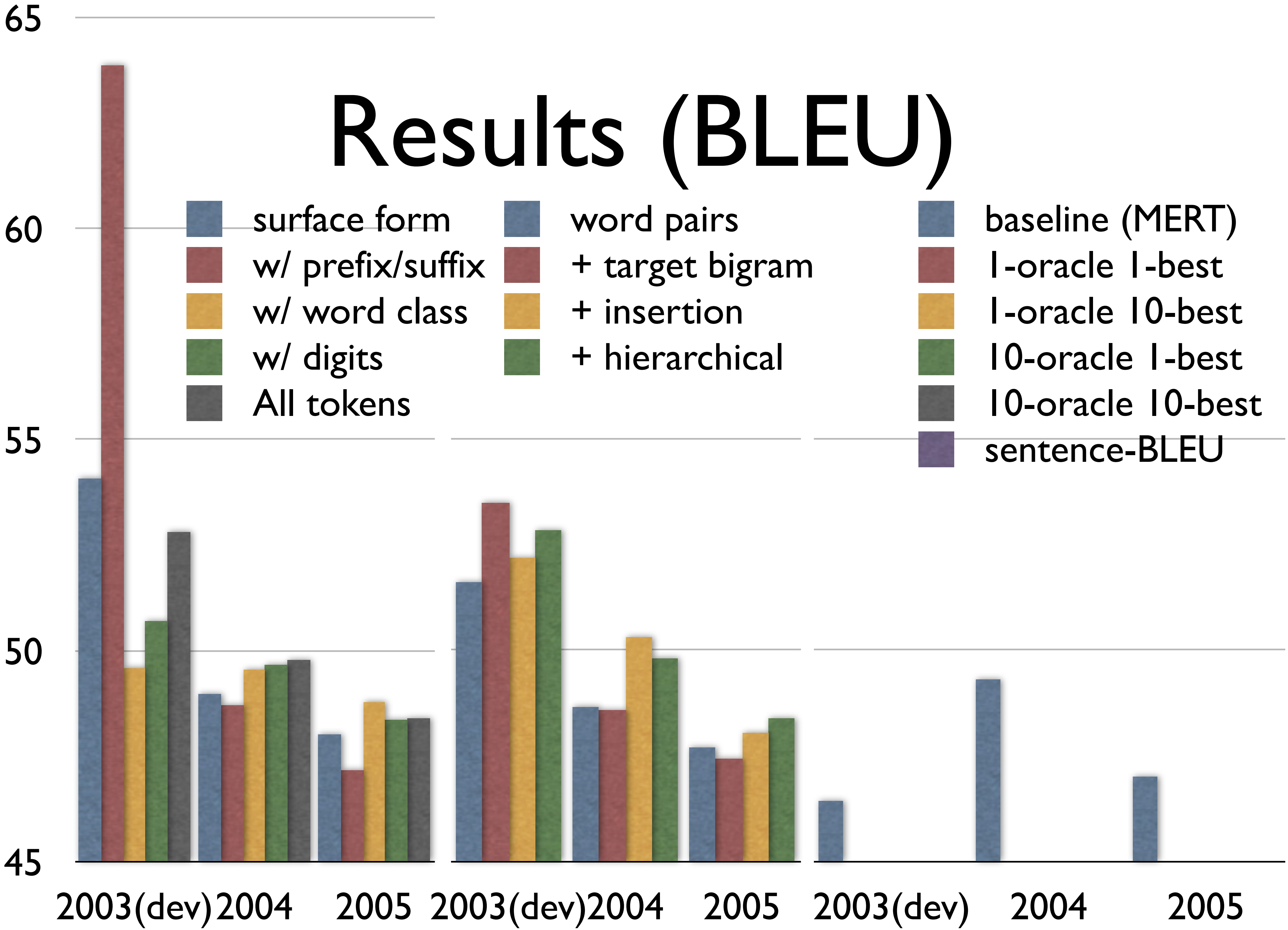
Results (BLEU)



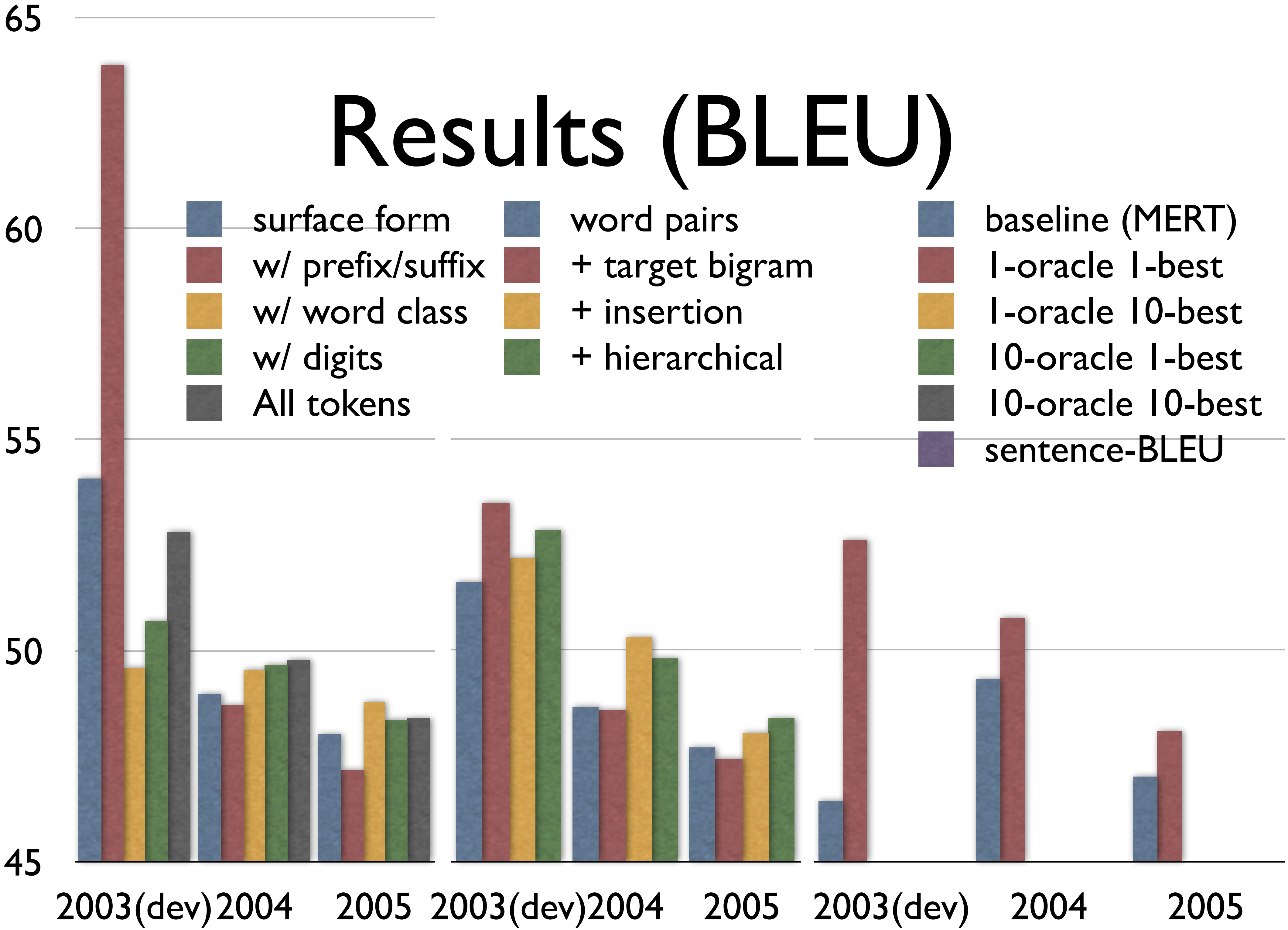
Results (BLEU)

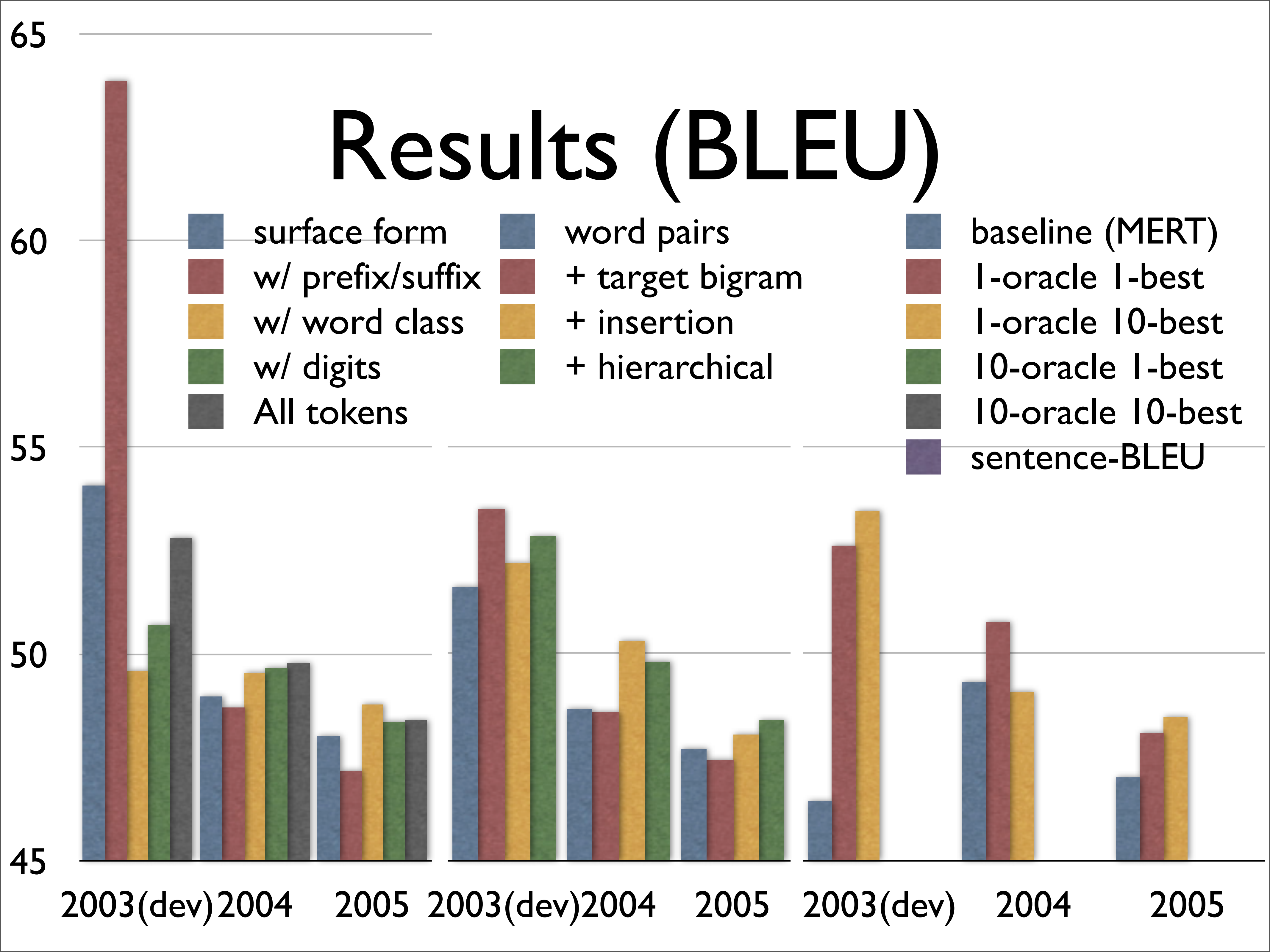


Results (BLEU)

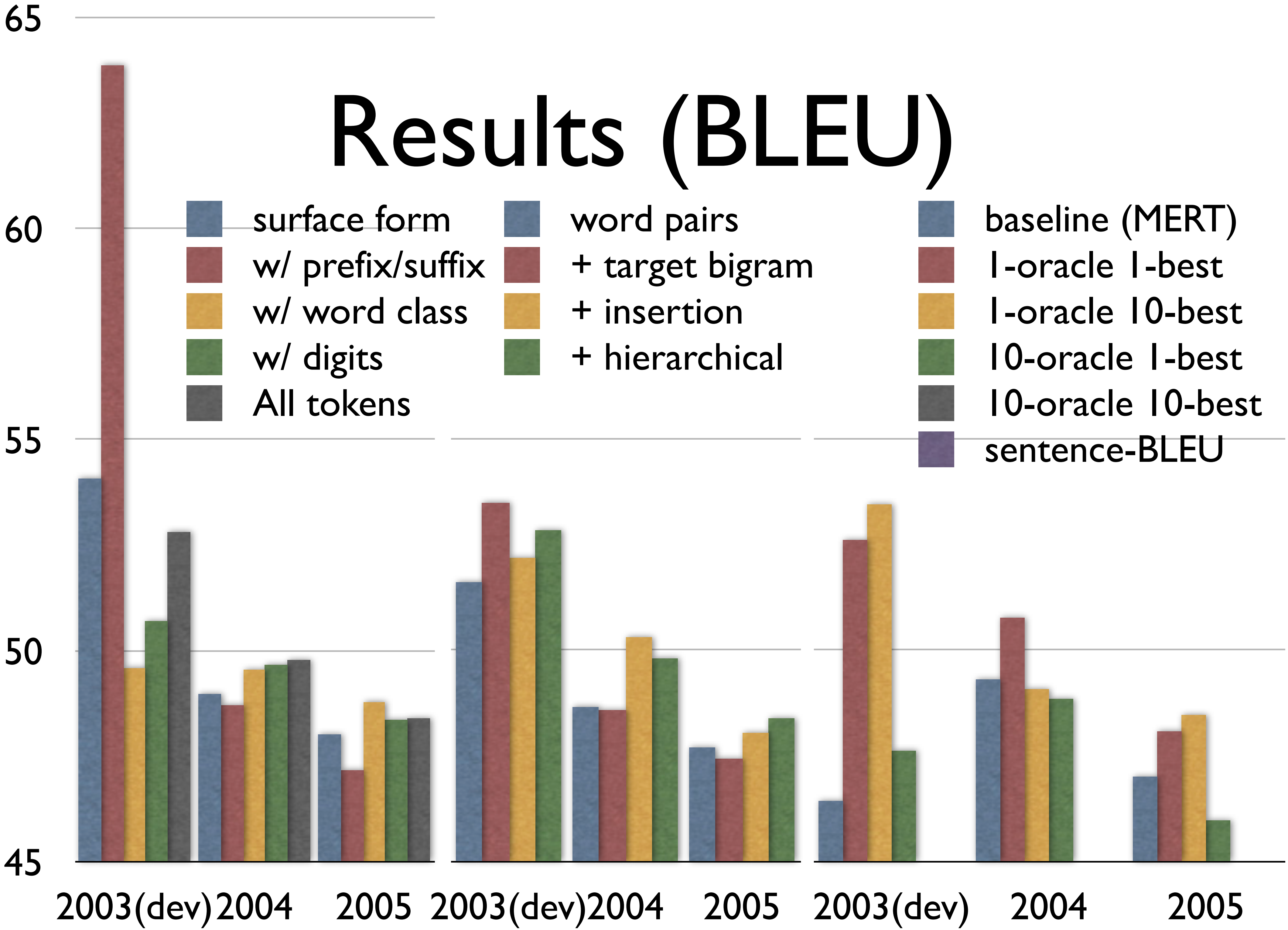


Results (BLEU)

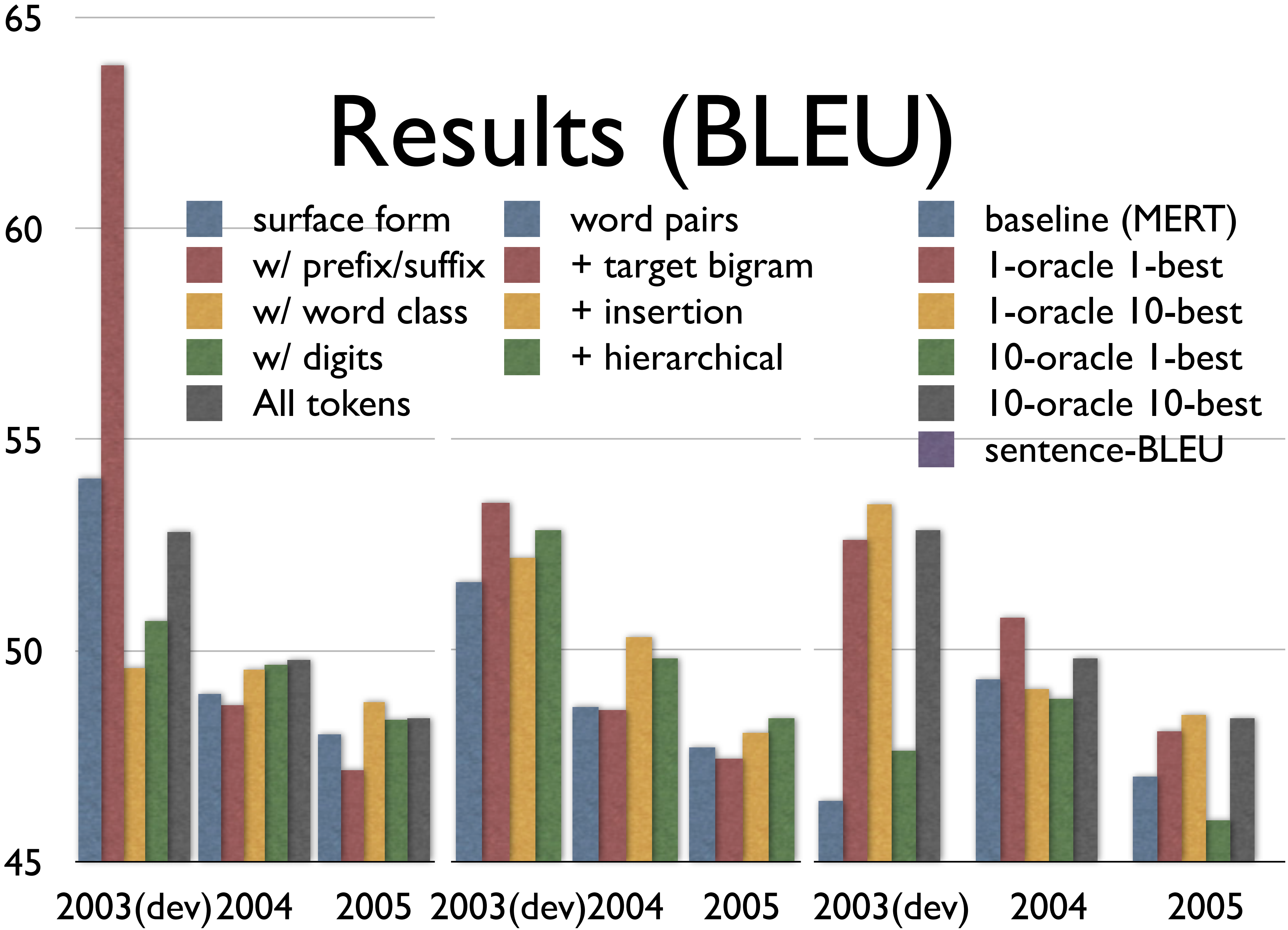



Results (BLEU)
surface form
w/ prefix/suffix
w/ word class
w/ digits
All tokens
word pairs
+ target bigram
+ insertion
+ hierarchical
baseline (MERT)
1-oracle 1-best
1-oracle 10-best
10-oracle 1-best
10-oracle 10-best
sentence-BLEU
65
60
55
50
45
2003(dev)
2004
2005
2003(dev)
2004
2005
2003(dev)
2004
2005

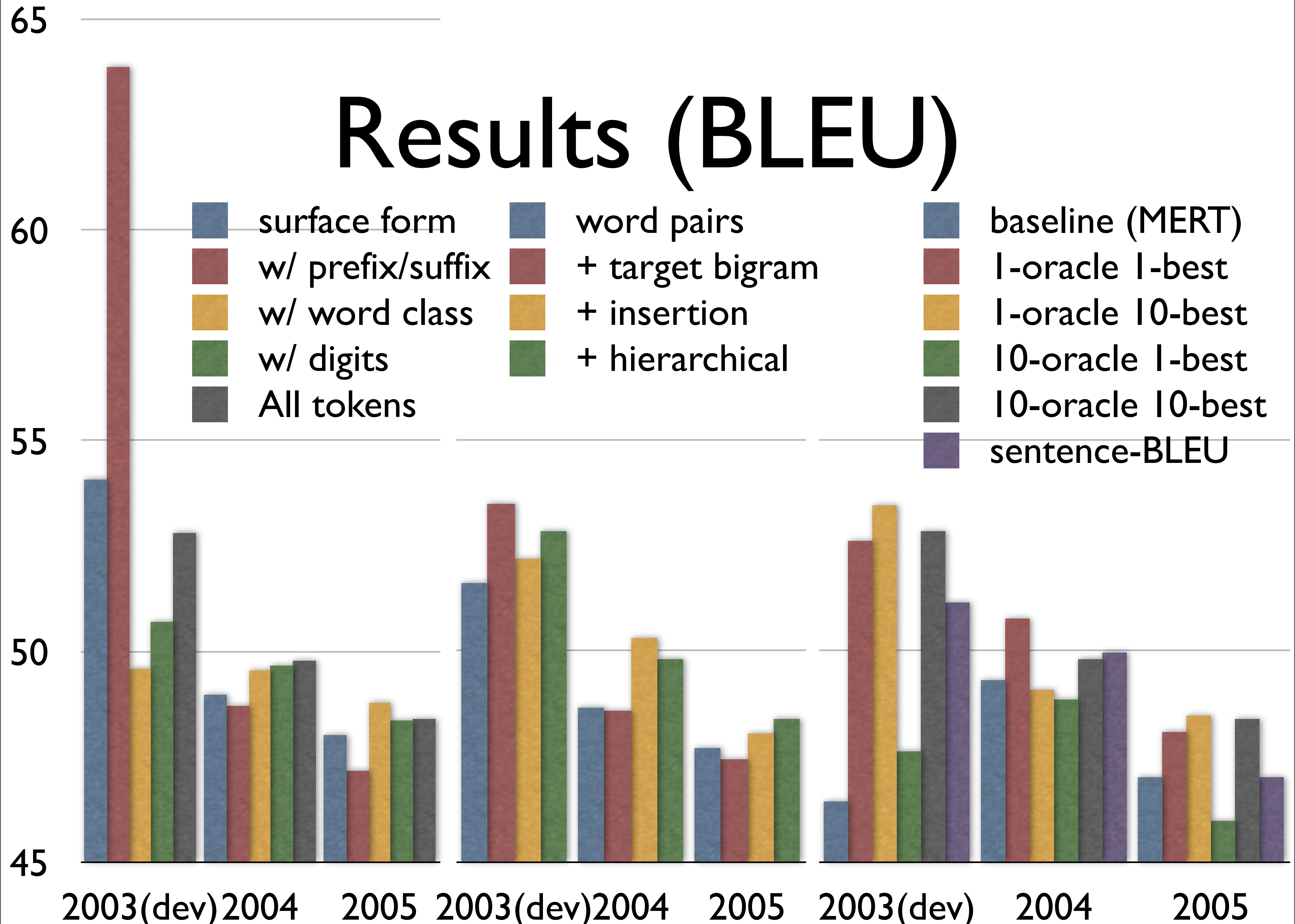
Results (BLEU)



Results (BLEU)



Results (BLEU)



Two-fold cross validation

	closed test		open test	
	NIST	BLEU	NIST	BLEU
baseline	10.71	44.79	10.68	44.44
online	11.58	53.42	10.90	47.64

Summary

- Online Large-Margin Training (This work)
  - Memorized local update strategy
  - Approximated BLEU
- SGD Training (Tillmann and Zhang, 2006)
  - Precomputed oracles/no real valued features.
- Perceptron Training (Liang et al., 2006)
  - Local update strategy

Conclusion

- Exploited only a small data set for millions of features:
    - Easy to explore alternative features, such as POS/NE etc.
- Future work:
    - Larger data + more features.