Integrated Supertagging and Parsing

Michael Auli
University of Edinburgh
Parsing

Marcel proved completeness
Marcel proved completeness
CCG Parsing

Combinatory Categorial Grammar (CCG; Steedman 2000)
Marcel proved completeness

\[
<\text{proved}, (S \backslash NP)/\text{NP}, \text{completeness}>\]
Marcel proved completeness.

CCG Parsing

<proved, (S\NP)/NP, completeness>

<proved, (S\NP)/NP, Marcel>
Why CCG Parsing?

- MT: Can analyse nearly any span in a sentence
  (Auli ’09; Mehay ‘10; Zhang & Clark 2011; Weese et. al. ’12)
  e.g. “conjectured and proved completeness” ⊢ S\NP

- Composition of regular and context-free languages --
  mirrors situation in syntactic MT (Auli & Lopez, ACL 2011)

- Transparent interface to semantics (Bos et al. 2004)
  e.g. proved ⊢ (S\NP)/NP : λx.λy.proved’ xy
CCG Parsing is hard!

Over 22 tags per word! (Clark & Curran 2004)
Supertagging

Marcel proved completeness
Supertagging

Marcel  proved  completeness

\[ NP \quad (S \backslash NP)/NP \quad NP \]
Supertagging

Marcel proved completeness

\[
\frac{NP}{NP} > \frac{(S\backslash NP)/NP}{NP} \quad S\backslash NP \quad S
\]
Supertagging

time  flies  like  an  arrow
NP   S\NP  (S\NP)/NP  NP/\NP  NP

Supertagging

- time: NP
- flies: S\NP
- like: (S\NP)/NP
- an: NP/NP
- arrow: NP
The Problem

- Supertagger has no sense of overall grammaticality.
- But parser restricted by its decisions.
- Supertagger probabilities not used in parser.
The Problem

- Supertagger has no sense of overall grammaticality.
- But parser restricted by its decisions.
- Supertagger probabilities not used in parser.
The Problem

- Supertagger has no sense of overall grammaticality.
- But parser restricted by its decisions.
- Supertagger probabilities not used in parser.
The Problem

- Supertagger has no sense of overall grammaticality.
- But parser restricted by its decisions.
- Supertagger probabilities not used in parser.
The Problem

- Supertagger has no sense of overall grammaticality.
- But parser restricted by its decisions.
- Supertagger probabilities not used in parser.
The Problem

- Supertagger has no sense of overall grammaticality.
- But parser restricted by its decisions.
- Supertagger probabilities not used in parser.
This talk

• Analysis of state-of-the-art approach
  Trade-off between efficiency and accuracy (ACL 2011a)

• Integrated supertagging and parsing
  with Loopy Belief Propagation and Dual Decomposition (ACL 2011b)

• Training the integrated model
  with Softmax-Margin towards task-specific metrics (EMNLP 2011)

Methods achieve most accurate CCG parsing results.
This talk

- Analysis of state-of-the-art approach
  Trade-off between efficiency and accuracy (ACL 2011a)

- Integrated supertagging and parsing
  with Loopy Belief Propagation and Dual Decomposition (ACL 2011b)

- Training the integrated model
  with Softmax-Margin towards task-specific metrics (EMNLP 2011)

Methods achieve
Adaptive Supertagging

time
\[ NP \]
flies
\[ S \setminus NP \]
like
\[ (S \setminus NP) / NP \]
an
\[ NP / NP \]
arrow
\[ NP \]
Adaptive Supertagging

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>flies</td>
<td>like</td>
<td>an</td>
<td>arrow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>$S\backslash NP$</td>
<td>$(S\backslash NP)/NP$</td>
<td>NP/NP</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP/NP</td>
<td>NP</td>
<td>....</td>
<td>NP/NP</td>
<td>NP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

((S\backslash NP)\backslash (S\backslash NP))/NP

....
Adaptive Supertagging

Adaptive Supertagging


- Algorithm:
  - Run supertagger.
  - Return tags with posterior higher than some alpha.
  - Parse by combining tags (CKY).
  - If parsing succeeds, stop.
  - If parsing fails, lower alpha and repeat.
Adaptive Supertagging


- Algorithm:
  - Run supertagger.
  - Return tags with posterior higher than some alpha.
  - Parse by combining tags (CKY).
  - If parsing succeeds, stop.
  - If parsing fails, lower alpha and repeat.

- Q: are parses returned in early rounds suboptimal?
Answer...

Oracle parsing
(Huang 2008)

Standard parsing
(Clark and Curran 2007)

Labelled F-score

100 

97 

95 

92 

Tight beam

Loose beam
Answer...

**Oracle parsing**  
(Huang 2008)

**Standard parsing**  
(Clark and Curran 2007)

![Diagram showing F-scores for oracle parsing and standard parsing. The graph compares tight and loose beams. The oracle parsing shows a higher F-score compared to standard parsing.](image-url)
Answer...

Oracle parsing
(Huang 2008)

Standard parsing
(Clark and Curran 2007)

Labelled F-score

Tight beam
Loose beam
Answer...

Oracle parsing
(Huang 2008)

<table>
<thead>
<tr>
<th>Labelled F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
</tr>
<tr>
<td>95</td>
</tr>
<tr>
<td>97</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Labelled F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>95</td>
</tr>
<tr>
<td>97</td>
</tr>
<tr>
<td>100</td>
</tr>
</tbody>
</table>

Standard parsing
(Clark and Curran 2007)

<table>
<thead>
<tr>
<th>Labelled F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
</tr>
<tr>
<td>87</td>
</tr>
<tr>
<td>88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Labelled F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
</tr>
<tr>
<td>87</td>
</tr>
<tr>
<td>90</td>
</tr>
</tbody>
</table>
**Parsing**

Note: only sentences parsable at all beam settings.

**Supertagger beam**

- Most aggressive
  - 0.075
- Most aggressive
  - 0.00001

**Model score**

- 85600
- 87400

**Labelled F-score**

- 88.2
- 89.8
A graph showing the model score and F-measure for different supertagger beam settings. The x-axis represents the supertagger beam settings, ranging from most aggressive (0.00001) to least aggressive (0.075). The y-axis shows the model score and F-measure, ranging from 85600 to 89.8.

Note: only sentences parsable at all beam settings.
Oracle Parsing

Note: only sentences parsable at all beam settings.
What’s happening here?
What’s happening here?

• Supertagger keeps parser from making serious errors.
What’s happening here?

- Supertagger keeps parser from making serious errors.
- But it also occasionally prunes away useful parses.
What's happening here?

- Supertagger keeps parser from making serious errors.
- But it also occasionally prunes away useful parses.
- Why not combine supertagger and parser into one?
Overview

• Analysis of state-of-the-art approach
  Trade-off between efficiency and accuracy (ACL 2011a)

• Integrated supertagging and parsing
  with Loopy Belief Propagation and Dual Decomposition (ACL 2011b)

• Training the integrated model
  with Softmax-Margin towards task-specific metrics (EMNLP 2011)
Overview

• Analysis of state-of-the-art approach
  Trade-off between efficiency and accuracy (ACL 2011a)

• Integrated supertagging and parsing
  with Loopy Belief Propagation and Dual Decomposition (ACL 2011b)

• Training the integrated model
  with Softmax-Margin towards task-specific metrics (EMNLP 2011)
Integrated Model

- Supertagger & parser are log-linear models.

- **Idea**: combine their features into one model.

- **Problem**: Exact computation of marginal or maximum quantities becomes very expensive because parsing and tagging submodels must agree on the tag sequence.
Integrated Model

• Supertagger & parser are log-linear models.

• **Idea**: combine their features into one model.

• **Problem**: Exact computation of marginal or maximum quantities becomes very expensive because parsing and tagging submodels must agree on the tag sequence.

original parsing problem: $B C \rightarrow A \quad O(Gn^3)$
Integrated Model

• Supertagger & parser are log-linear models.

• Idea: combine their features into one model.

• Problem: Exact computation of marginal or maximum quantities becomes very expensive because parsing and tagging submodels must agree on the tag sequence.

original parsing problem: \[ B \rightarrow C \rightarrow A \quad O(Gn^3) \]

new parsing problem: \[ qB_s sC_r \rightarrow qA_r \quad O(G^3n^3) \]
Integrated Model

- Supertagger & parser are log-linear models.

- **Idea**: combine their features into one model.

- **Problem**: Exact computation of marginal or maximum quantities becomes very expensive because parsing and tagging submodels must agree on the tag sequence.

original parsing problem: \[ B \ C \rightarrow A \quad O(Gn^3) \]

new parsing problem: \[ qB_{s\ s}C_r \rightarrow qA_r \quad O(G^3n^3) \]

Intersection of a regular and context-free language

(Bar-Hillel et al. 1964)
Approximate Algorithms

- Loopy belief propagation: approximate calculation of marginals. (Pearl 1988; Smith & Eisner 2008)

- Dual decomposition: exact (sometimes) calculation of maximum. (Dantzig & Wolfe 1960; Komodakis et al. 2007; Koo et al. 2010)
Belief Propagation
Belief Propagation

Forward-backward is belief propagation (Smyth et al. 1997)
Belief Propagation

Forward-backward is belief propagation (Smyth et al. 1997)
Belief Propagation

emission message: $e_{i,j}$

forward message: $f_{i,j} = \sum_{j'} f_{i-1,j'} e_{i-1,j'} t_{j',j}$

backward message: $b_{i,j} = \sum_{j'} b_{i+1,j'} e_{i+1,j'} t_{j,j'}$

belief (probability) that tag $j$ is at position $i$: $p_{i,j} = \frac{1}{Z} f_{i,j} e_{i,j} b_{i,j}$

Forward-backward is belief propagation (Smyth et al. 1997)
Belief Propagation

Notational convenience: one factor describes whole distribution over supertag sequence...

Marcel proved completeness
Belief Propagation

We can also do the same for the distribution over parse trees.
Belief Propagation

We can also do the same for the distribution over parse trees! (Case-factor diagrams: McAllester et al. 2008)

Inside-outside is belief propagation (Sato 2007)
Belief Propagation

Marcel proved completeness.
Belief Propagation

Graph is not a tree!

$parsing$ factor

Marcel proved completeness

$supertagging$ factor
Loopy Belief Propagation

Graph is not a tree!

parsing factor

supertagging factor

Marcel  proved  completeness
\textit{Loopy} Belief Propagation

Graph is not a tree!

\textit{parsing} factor

\textit{supertagging} factor

Marcel \quad proved \quad \text{forward-backward completenes}
Loopy Belief Propagation

Graph is not a tree!

parsing factor

supertagging factor

Marcel

proved

forward-backward

completeness

inside-outside
Loopy Belief Propagation

Marcel proved completeness.

Graph is not a tree!

parsing factor

supertagging factor

Marcel proved completeness

forward-backward

inside-outside
Loopy Belief Propagation

Graph is not a tree!

\[ p_{i,j} = \frac{1}{Z} f_{i,j} e_{i,j} b_{i,j} o_{i,j} \]

- parsing factor
- supertagging factor
- Marcel
- proved
- completeness
- forward-backward
- inside-outside
Loopy Belief Propagation

Graph is not a tree!

\[ p_{i,j} = \frac{1}{Z} f_{i,j} e_{i,j} b_{i,j} o_{i,j} \]

- Parsing factor
- Supertagging factor
- Marcel
- proved
- completeness
- forward-backward
- inside-outside

Marcel proved completeness.

Loopy Belief Propagation
Loopy Belief Propagation

- Computes approximate marginals, no guarantees.
- Complexity is additive: $O(Gn^3 + Gn)$
- Used to compute minimum-risk parse (Goodman 1996).
Marcel proved completeness.
Dual Decomposition

Marcel proved completeness

parsing factor
\( f(y) \)

supertagging factor
\( g(z) \)

Marcel \quad \text{proved} \quad \text{completeness}
Dual Decomposition

\[
\arg \max_{y, z} f(y) + g(z) \quad \text{s.t. } y(i, t) = z(i, t) \text{ for all } i, t
\]
Dual Decomposition

\[
\arg \max_{y,z} f(y) + g(z) \quad \text{s.t. } y(i, t) = z(i, t) \text{ for all } i, t
\]
Dual Decomposition

\[
\text{arg max}_{y, z} f(y) + g(z) \quad \text{s.t. } y(i, t) = z(i, t) \text{ for all } i, t
\]

\[
L(u) = \max_y f(y) + \sum_{i, t} u(i, t) \cdot y(i, t)
\]

\[
+ \max_z g(z) - \sum_{i, t} u(i, t) \cdot z(i, t)
\]
Dual Decomposition

\[ \arg \max_{y, z} f(y) + g(z) \quad \text{s.t. } y(i, t) = z(i, t) \text{ for all } i, t \]

\[ L(u) = \max_y f(y) + \sum_{i, t} u(i, t) \cdot y(i, t) + \max_z g(z) - \sum_{i, t} u(i, t) \cdot z(i, t) \]
Dual Decomposition

\[
\arg \max_{y,z} f(y) + g(z) \quad \text{s.t.} \quad y(i, t) = z(i, t) \quad \text{for all} \quad i, t
\]

\[
L(u) = \max_y f(y) + \sum_{i,t} u(i, t) \cdot y(i, t) + \max_z g(z) - \sum_{i,t} u(i, t) \cdot z(i, t)
\]
Dual Decomposition

$$\arg \max_{y, z} f(y) + g(z) \quad \text{s.t. } y(i, t) = z(i, t) \text{ for all } i, t$$

$$L(u) = \max_y f(y) + \sum_{i, t} u(i, t) \cdot y(i, t)$$

$$+ \max_z g(z) - \sum_{i, t} u(i, t) \cdot z(i, t)$$

Dual objective: find assignment of $u(i, t)$ that minimises $L(u)$
Dual Decomposition

\[ \text{arg max}_{y,z} f(y) + g(z) \quad \text{s.t. } y(i, t) = z(i, t) \text{ for all } i, t \]

\[ L(u) = \max_y f(y) + \sum_{i,t} u(i, t) \cdot y(i, t) \]

\[ + \max_z g(z) - \sum_{i,t} u(i, t) \cdot z(i, t) \]

Dual objective: find assignment of \( u(i, t) \) that minimises \( L(u) \)

\[ u(i, t) = u(i, t) + \alpha \cdot [y(i, t) - z(i, t)] \quad \text{(Rush et al. 2010)} \]

Solution provably solves original problem.
Dual Decomposition

Marcel proved completeness
Dual Decomposition

Marcel proved completeness.
Dual Decomposition

Marcel proved completeness

"Message passing" (Komodakis et al. 2007)

parsing factor

Viterbi parse

supertagging factor

Viterbi tags

Marcel -> proved -> completeness

Message passing
Dual Decomposition

- Computes *exact* maximum, *if* it converges.
  - Otherwise: return best parse seen (approximation).
- Complexity is additive: $O(Gn^3 + Gn)$
- Use to compute Viterbi solutions.
Experiments

• Standard parsing task:
  • C&C Parser and supertagger *(Clark & Curran 2007)*.
  • CCGBank standard train/dev/test splits.
  • Piecewise optimisation *(Sutton and McCallum 2005)*
  • Approximate algorithms used to decode test set.
Experiments: Accuracy over time
Experiments: Accuracy over time

- Tight search (AST)
- Loose search (Rev)

Graph showing labelled F-score over iterations with different search methods (BL AST, BP AST, DD AST, BL Rev, BP Rev, DD Rev).
Experiments: Convergence

![Convergence Graph]

- BP AST
- BP Reverse
- DD AST
- DD Reverse

Convergence rate (%) vs. Iterations
Experiments: Convergence

Dual decomposition exact in 99.7% of cases
What about belief propagation?
Experiments: BP Exactness
Experiments: BP Exactness

![Graph showing match (%) vs iterations for BP exactness experiments. The graph shows a trend where match percentage increases with iterations and stabilizes above 92% for k=1000.]
Experiments: BP Exactness

Instantly, 91% match final DD solutions!
Takes DD 15 iterations to reach same level.
Experiments: Accuracy

Test set results

<table>
<thead>
<tr>
<th>Labelled F-measure</th>
<th>Tight beam</th>
<th>Loose beam</th>
</tr>
</thead>
<tbody>
<tr>
<td>89</td>
<td>Baseline</td>
<td>Belief Propagation</td>
</tr>
<tr>
<td>88.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiments: Accuracy

Test set results

Labelled F-measure

<table>
<thead>
<tr>
<th></th>
<th>Tight beam</th>
<th>Loose beam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>87.7</td>
<td>88.1</td>
</tr>
<tr>
<td>Belief Propagation</td>
<td>88.3</td>
<td>89.0</td>
</tr>
<tr>
<td>Dual Decomposition</td>
<td>87.7</td>
<td>89.0</td>
</tr>
</tbody>
</table>
Experiments: Accuracy

Test set results

<table>
<thead>
<tr>
<th>Beam</th>
<th>Baseline</th>
<th>Belief Propagation</th>
<th>Dual Decomposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tight beam</td>
<td>87.7</td>
<td>88.3</td>
<td>88.1</td>
</tr>
<tr>
<td>Loose beam</td>
<td>87.7</td>
<td>88.9</td>
<td>88.8</td>
</tr>
</tbody>
</table>

+1.1

Note: BP accuracy after 1 iteration; DD accuracy after 25 iterations
Experiments: Accuracy

Test set results

![Bar chart showing accuracy for Tight beam and Loose beam.]

- **Baseline**: 87.7, 87.7
- **Belief Propagation**: 88.3, 88.9
- **Dual Decomposition**: 88.1, 88.8

The best published result is +1.1 higher than the baseline.

**Note**: BP accuracy after 1 iteration; DD accuracy after 25 iterations.
Oracle Results Again

Belief Propagation

Dual Decomposition
Summary so far

- Supertagging efficiency comes at the cost of accuracy.
- Interaction between parser and supertagger can be exploited in an integrated model.
- Practical inference for complex integrated model.
- First empirical comparison between dual decomposition and belief propagation on NLP task.
- Loopy belief propagation is fast, accurate and exact.
Overview

• Analysis of state-of-the-art approach
  Trade-off between efficiency and accuracy (ACL 2011a)

• Integrated supertagging and parsing
  with Loopy Belief Propagation and Dual Decomposition (ACL 2011b)

• Training the integrated model
  with Softmax-Margin towards task-specific metrics (EMNLP 2011)
Overview

- Analysis of state-of-the-art approach
  Trade-off between efficiency and accuracy (ACL 2011a)

- Integrated supertagging and parsing
  with Loopy Belief Propagation and Dual Decomposition (ACL 2011b)

- Training the integrated model
  with Softmax-Margin towards task-specific metrics (EMNLP 2011)
Training the Integrated Model

- So far optimised Conditional Log-Likelihood (CLL).

- Optimise towards task-specific metric e.g. $F_1$ such as in SMT (Och, 2003).

- Past work used approximations to Precision (Taskar et al. 2004).

- Contribution: Do it exactly and verify approximations.
Marcel proved completeness

CCG: Labelled, directed dependency recovery
(Clark & Hockenmaier, 2002)

\[
\begin{array}{c}
\text{Marcel} \\
\text{proved} \\
\text{completeness} \\
\text{NP} \\
(S \backslash \text{NP}) / \text{NP} \\
S \backslash \text{NP} \\
S \\
\end{array}
\]

\[
\langle \text{proved, } (S \backslash \text{NP}) / \text{NP}, \text{completeness} \rangle
\]

\[
\langle \text{proved, } (S \backslash \text{NP}) / \text{NP}, \text{Marcel} \rangle
\]
Evaluate this

Marcel proved completeness

CCG: Labelled, directed dependency recovery
(Clark & Hockenmaier, 2002)

<proved, (S\NP)/NP, completeness>
<proved, (S\NP)/NP, Marcel>
<proved, (S\NP)/NP, completeness>
<proved, (S\NP)/NP, Marcel>
Parsing Metrics

\( y = \text{dependencies in ground truth} \)
\( y' = \text{dependencies in proposed output} \)

\[ |y \cap y'| = n \quad \text{correct dependencies returned} \]
\[ |y'| = d \quad \text{all dependencies returned} \]
Parsing Metrics

\( y = \) dependencies in ground truth
\( y' = \) dependencies in proposed output

\(|y \cap y'| = n\) correct dependencies returned
\(|y'| = d\) all dependencies returned

Precision
\[ P(y, y') = \frac{|y \cap y'|}{|y'|} = \frac{n}{d} \]

Recall
\[ R(y, y') = \frac{|y \cap y'|}{|y|} = \frac{n}{|y|} \]

F-measure
\[ F_1(y, y') = \frac{2PR}{P + R} = \frac{2|y \cap y'|}{|y| + |y'|} = \frac{2n}{d + |y|} \]
Softmax-Margin Training

(Sha & Saul, 2006; Povey & Woodland, 2008; Gimpel & Smith, 2010)

- Discriminative.
- Probabilistic.
- Convex objective.
- Minimises bound on expected risk for a given loss function.
- Requires little change to existing CLL implementation.
Softmax-Margin Training

CLL: \[
\min_{\theta} \sum_{i=1}^{m} \left[ -\theta^T f(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, y)\} \right]
\]
Softmax-Margin Training

\[
\text{CLL:} \quad \min_{\theta} \sum_{i=1}^{m} \left[ -\theta^T f(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, y)\} \right]
\]
Softmax-Margin Training

\[ \min_\theta \sum_{i=1}^{m} \left[ -\theta^T f(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, y)\} \right] \]

\[ \min_\theta \sum_{i=1}^{m} \left[ -\theta^T f(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, y) + \ell(y^{(i)}, y)\} \right] \]
Softmax-Margin Training

CLL: \[ \min_\theta \sum_{i=1}^{m} \left[ -\theta^T f(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, y)\} \right] \]

SMM: \[ \min_\theta \sum_{i=1}^{m} \left[ -\theta^T f(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, y) + \ell(y^{(i)}, y)\} \right] \]
Softmax-Margin Training

Contrary to the traditional maximum margin training, the Softmax-Margin Training can be divided into two parts:

**CLL:**
\[
\min_{\theta} \sum_{i=1}^{m} \left[ -\theta^T f(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, y)\} \right]
\]

**SMM:**
\[
\min_{\theta} \sum_{i=1}^{m} \left[ -\theta^T f(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\theta^T f(x^{(i)}, y) + \ell(y^{(i)}, y)\} \right]
\]

- Penalise high-loss outputs.
- Re-weight outcomes by loss function.
- Loss function an unweighted feature -- if decomposable.
Decomposability

- CKY assumes weights factor over substructures (node + children = substructure).
- A *decomposable* loss function must factor identically.
Decomposability

- CKY assumes weights factor over substructures (node + children = substructure).
- A decomposable loss function must factor identically.

Marcel proved completeness.

\[ S_{0,3} \]
\[ NP_{0,1} \]
\[ (S \setminus NP)/NP_{1,2} \]
\[ S \setminus NP_{1,3} \]
\[ NP_{2,3} \]

Marcel proved completeness.
Decomposability

- CKY assumes weights factor over substructures (node + children = substructure).
- A *decomposable* loss function must factor identically.

Marcel proved completeness.
Decomposability

\[ |y \cap y'| = n \quad \text{correct dependencies returned} \]
\[ |y'| = d \quad \text{all dependencies returned} \]

\[ n = n_1 + n_2 \]
Decomposability

Correct dependency counts

\[ |y \cap y'| = n \quad \text{correct dependencies returned} \]
\[ |y'| = d \quad \text{all dependencies returned} \]

\[ n = n_1 + n_2 \]
Decomposability

F-measure

\[ |y \cap y'| = n \quad \text{correct dependencies returned} \]
\[ |y'| = d \quad \text{all dependencies returned} \]
Decomposability

F-measure

\[ |y \cap y'| = n \quad \text{correct dependencies returned} \]
\[ |y'| = d \quad \text{all dependencies returned} \]

\[ f = f_1 \otimes f_2 \]

\( S_{0,3}: f \)

\( S \setminus NP_{1,3}: f_2 \)

\( NP_{0,1}: f_1 \)

Marcel

proved

completeness

\((S \setminus NP)/NP_{1,2}\)

\( NP_{2,3} \)
Decomposability

F-measure

\[ |y \cap y'| = n \quad \text{correct dependencies returned} \]
\[ |y'| = d \quad \text{all dependencies returned} \]

Marcel proved completeness

\( f = f_1 \otimes f_2 \)

Approximations!
Approximate Loss Functions

Marcel proved completeness.
Approximate Loss Functions

for each substructure:

\( n_+ \)  correct dependencies
\( d_+ \)  all dependencies
\( c_+ \)  gold dependencies

\( S_{0,3}:1,1 \)
\( NP_{0,1}:0,0 \)
\( S\setminus NP_{1,3}:1,1 \)
\( (S\setminus NP)/NP_{1,2}:0,0 \)
\( NP_{2,3}:0,0 \)

Marcel  proved  completeness
Approximate Loss Functions

for each substructure:

- \( n_+ \): correct dependencies
- \( d_+ \): all dependencies
- \( c_+ \): gold dependencies

\[
\begin{align*}
\text{DecP}(y) &= \sum_{t \in T(y)} d_+(t) - n_+(t) \\
\text{DecR}(y) &= \sum_{t \in T(y)} c_+(t) - n_+(t) \\
\text{DecF1}(y) &= \text{DecP}(y) + \text{DecR}(y)
\end{align*}
\]
Approximate Losses with CKY

items $A_{i,j}$

target analysis
correct dependencies
all dependencies

time$_1$ flies$_2$ like$_3$ an$_4$ arrow$_5$
Approximate Losses with CKY

items $A_{i,j}$

target analysis
correct dependencies
all dependencies

$NP_{0,1}$
$S \setminus NP_{1,2}$
$((S \setminus NP) \setminus (S \setminus NP))/NP_{2,3}$

$NP_{3,4}$
$NP_{4,5}$

$\text{time}_1$
$\text{flies}_2$
$\text{like}_3$
$\text{an}_4$
$\text{arrow}_5$
Approximate Losses with CKY

items $A_{i,j}$

target analysis

correct dependencies

all dependencies
Approximate Losses with CKY

items $A_{i,j}$

target analysis

correct dependencies

all dependencies

($S \backslash NP) \backslash (S \backslash NP)_{2,5}$

DecF$_1$(1,1)

$((S \backslash NP) \backslash (S \backslash NP))/NP_{2,3}$

NP$_3,5$

DecF$_1$(1,1)

NP$_{3,4}$

NP$_{4,5}$

NP$_{0,1}$

S \backslash NP$_{1,2}$

time$_1$

flies$_2$

like$_3$

NP$_{4,5}$

an$_4$
Approximate Losses with CKY

items $A_{i,j}$

target analysis
correct dependencies
all dependencies

$S \setminus NP_{1,5}$

DecF$_1(1,1)$

$((S \setminus NP)/(S \setminus NP)) / NP_{2,3}$

DecF$_1(1,1)$

$NP_{3,5}$

DecF$_1(1,1)$

NP/ NP$_{3,4}$

NP$_{4,5}$
Approximate Losses with CKY

items $A_{i,j}$

$S_{0,5}$

DecF$_1(1,1)$

$S \backslash NP_{1,5}$

DecF$_1(1,1)$

$(S \backslash NP) \backslash (S \backslash NP)_{2,5}$

DecF$_1(1,1)$

$((S \backslash NP) \backslash (S \backslash NP)) / NP_{2,3}$

DecF$_1(1,1)$

NP$/NP_{3,4}$

NP$_{3,5}$

DecF$_1(1,1)$

NP$_{4,5}$

NP$_{4,5}$

target analysis correct dependencies all dependencies
Approximate Losses with CKY

items $A_{i,j}$

target analysis
correct dependencies
all dependencies
Approximate Losses with CKY

items $A_{i,j}$

another analysis
correct dependencies
all dependencies

time$_1$ flies$_2$ like$_3$ an$_4$ arrow$_5$
Approximate Losses with CKY

items $A_{i,j}$

another analysis
correct dependencies
all dependencies

$NP/NP_{0,1}$ $NP_{0,1}$

time$_1$ flies$_2$

$(S\setminus NP)/NP_{2,3}$

like$_3$

$NP/NP_{3,4}$

an$_4$

$NP_{4,5}$

arrow$_5$
Approximate Losses with CKY

items $A_{i,j}$

another analysis
correct dependencies
all dependencies

$NP/NP_{0,1}$

$NP_{0,1}$

time$_1$

flies$_2$

$NP/NP_{0,1}$

$(S/NP)/NP_{2,3}$

$NP_{3,5}$

DecF$_1(1,1)$

$NP/NP_{3,4}$

an$_4$

arrow$_5$
Approximate Losses with CKY

items $A_{i,j}$

another analysis

correct dependencies

all dependencies

Approximate Losses with CKY
Approximate Losses with CKY

items $A_{i,j}$

another analysis
correct dependencies
all dependencies

Approximate Losses with CKY
Approximate Losses with CKY

items $A_{i,j}$

another analysis
correct dependencies
all dependencies

Approximate Losses with CKY
Approximate Losses with CKY

items $A_{i,j}$

another analysis

correct dependencies

all dependencies

Approximate Losses with CKY
Approximate Losses with CKY

items $A_{i,j}$

both analysis
correct dependencies
all dependencies
Approximate Losses with CKY

items $A_{i,j}$

both analysis correct dependencies all dependencies

Approximate Losses with CKY
Decomposability Revisited

F-measure

\[
\begin{align*}
|y \cap y'| &= n \quad \text{correct dependencies returned} \\
|y'| &= d \quad \text{all dependencies returned}
\end{align*}
\]

\[
F_1(y, y') = \frac{2n}{d + |y|}
\]

\[
\begin{align*}
S & \quad \text{S} \\
S_{0,3} & \quad \text{S}_{0,3} \\
\text{NP}_{0,1} & \quad \text{NP}_{0,1} \\
\text{S} \setminus \text{NP}_{1,3} & \quad \text{S} \setminus \text{NP}_{1,3} \\
(S \setminus \text{NP})/\text{NP}_{1,2} & \quad (S \setminus \text{NP})/\text{NP}_{1,2} \\
\text{NP}_{2,3} & \quad \text{NP}_{2,3}
\end{align*}
\]

Marcel proved completeness
Decomposability Revisited

F-measure

\[ |y \cap y'| = n \quad \text{correct dependencies returned} \]
\[ |y'| = d \quad \text{all dependencies returned} \]

\[ F_1(y, y') = \frac{2n}{d + |y|} \]

Diagram:
- \( S_{0,3} \)
- \( NP_{0,1}: n_1, d_1 \)
- \( S \setminus NP_{1,3}: n_2, d_2 \)
- \( (S \setminus NP)/NP_{1,2} \)
- \( NP_{2,3} \)
- Marcel
- proved
- completeness
Decomposability Revisited

F-measure

\[ |y \cap y'| = n \text{ correct dependencies returned} \]
\[ |y'| = d \text{ all dependencies returned} \]

\[ F_1(y, y') = \frac{2n}{d + |y|} \]

\[ f = \frac{2n_1}{d_1 + |y|} \otimes \frac{2n_2}{d_2 + |y|} \]

\[ = \frac{2(n_1 + n_2)}{d_1 + d_2 + 2|y|} \]

Marcel proved completeness
Exact Loss Functions

Marcel proved completeness.
Exact Loss Functions

- Treat sentence-level $F_1$ as non-local feature dependent on $n$, $d$. 

```
<table>
<thead>
<tr>
<th>Marcel</th>
<th>proved</th>
<th>completeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP$_{0,1}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S$_{0,3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP$_{1,2}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(S \ NP) / NP$_{1,2}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S \ NP$_{1,3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP$_{2,3}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Exact Loss Functions

- Treat sentence-level $F_1$ as *non-local feature* dependent on $n, d$.
- Result: new dynamic program over items $A_{i,j,n,d}$
Exact Losses with State-Split CKY

items $A_{i,j,n,d}$

correct dependencies
all dependencies

time$_1$ flies$_2$ like$_3$ an$_4$ arrow$_5$
Exact Losses with State-Split CKY

items $A_{i,j,n,d}$

correct dependencies
all dependencies

$S_{0,5,4,4}$

$S \setminus NP_{1,5,3,3}$

$(S \setminus NP) \setminus (S \setminus NP)_{2,5,2,2}$

$((S \setminus NP) \setminus (S \setminus NP)) / NP_{2,3,0,0}$

NP/NP$_{3,4,0,0}$

NP/NP$_{4,5,0,0}$

NP$_{3,5,1,1}$

NP$_{4,5,0,0}$

$(S \setminus NP) / NP_{2,3,0,0}$

$S \setminus NP_{1,2,0,0}$

$S_{0,5,4,4}$

NP$_{0,1,0,0}$

time$_1$

flies$_2$

like$_3$

an$_4$

arrow$_5$
Exact Losses with State-Split CKY

items $A_{i,j,n,d}$

correct dependencies
all dependencies

$S_{0,5,4,4}$

$NP_{0,1,0,0}$

$S \setminus NP_{1,2,0,0}$

$((S \setminus NP) \setminus (S \setminus NP))/NP_{2,3,0,0}$

$(S \setminus NP)/(S \setminus NP)_{2,5,2,2}$

$NP_{3,5,1,1}$

$NP_{4,5,0,0}$

$NP/NP_{3,4,0,0}$

$NP$
Exact Losses with State-Split CKY

items \( A_{i,j,n,d} \)

graph

\( F_1(4,4) \)
Exact Losses with State-Split CKY

items $A_{i,j,n,d}$

correct dependencies

all dependencies

$F_1(4,4)$

$F_1(1,4)$

$S_{0,5,4,4}$

$S_{0,5,1,4}$

$S \setminus NP_{1,5,3,3}$

$S \setminus NP_{3,5,1,2}$

$(S \setminus NP) \setminus (S \setminus NP)_{2,5,2,2}$

$(S \setminus NP) / NP_{2,3,0,0}$

$NP_{3,5,1,1}$

$NP_{4,5,0,0}$

$NP_{0,2,0,1}$

$NP_{0,1,0,0}$

$NP_{0,1,0,0}$

$time_1$

$flies_2$

$like_3$

$an_4$

$arrow_5$
Exact Losses with State-Split CKY

items $A_{i,j,n,d}$

correct dependencies
all dependencies

time
flies
like
an
arrow
Exact Losses with State-Split CKY

items $A_{i,j,n,d}$

$F_1(4,4)$

GOAL

$F_1(1,4)$

Speed $O(L^7)$

Space $O(L^4)$

$S_{0,5,4,4}$

$S_{0,5,1,4}$

$NP_{0,2,0,1}$

$NP/NP_{0,1,0,0}$

$NP_{0,1,0,0}$

$S_{NP_{1,2,0,0}}$

$((S\backslash NP)\backslash(S\backslash NP))_{2,5,2,2}$

$NP_{0,1,0,0}$

$S_{NP_{1,5,3,3}}$

$(S\backslash NP)/(S\backslash NP)_{2,3,0,0}$

$NP/\backslash NP_{3,4,0,0}$

$NP_{3,5,1,1}$

$NP_{4,5,0,0}$

$NP_{3,5,1,2}$

$NP_{3,5,1,1}$

items $A_{i,j,n,d}$

$NP_{0,2,0,1}$

$NP_{0,1,0,0}$

$S_{NP_{1,2,0,0}}$

$((S\backslash NP)\backslash(S\backslash NP))_{2,5,2,2}$

$NP_{0,1,0,0}$

$S_{NP_{1,5,3,3}}$

$(S\backslash NP)/(S\backslash NP)_{2,3,0,0}$

$NP/\backslash NP_{3,4,0,0}$

$NP_{3,5,1,1}$

$NP_{4,5,0,0}$
Exact Losses with State-Split CKY

in practice
48 x larger DP
30 x slower
Experiments

- Standard parsing task:
  - C&C Parser and supertagger (Clark & Curran 2007).
  - CCGBank standard train/dev/test splits.
  - Piecewise optimisation (Sutton and McCallum 2005)
Exact versus Approximate

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>88.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>87.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>86.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Approximate**
- **Exact**
Exact versus Approximate

<table>
<thead>
<tr>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximate</td>
<td>88.0</td>
<td></td>
</tr>
<tr>
<td>Exact</td>
<td>87.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>87.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>87.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>87.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>86.9</td>
<td></td>
</tr>
</tbody>
</table>
Exact versus Approximate

- Precision: Approximate (88.0) > Exact (87.6)
- Recall: Approximate (87.8) > Exact (87.1)
- F-measure: Approximate (88.0) > Exact (87.6)

Diagram shows bar charts for Precision, Recall, and F-measure comparing Approximate and Exact methods.
Exact versus Approximate

<table>
<thead>
<tr>
<th>Metric</th>
<th>Approximate</th>
<th>Exact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>88.0</td>
<td></td>
</tr>
<tr>
<td>Recall</td>
<td>87.3</td>
<td></td>
</tr>
<tr>
<td>F-measure</td>
<td>87.8</td>
<td></td>
</tr>
</tbody>
</table>
Exact versus Approximate

Approximate loss functions work, and much faster!
Softmax-Margin beats CLL

Test set results

<table>
<thead>
<tr>
<th>Labelled F-measure</th>
<th>Tight beam</th>
<th>Loose beam</th>
</tr>
</thead>
<tbody>
<tr>
<td>87.5</td>
<td>C&amp;C ‘07</td>
<td>DecF1</td>
</tr>
<tr>
<td>87.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>88.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>88.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Softmax-Margin beats CLL

Test set results

<table>
<thead>
<tr>
<th>Beam Type</th>
<th>C&amp;C '07</th>
<th>DecF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tight</td>
<td>87.7</td>
<td>88.1</td>
</tr>
<tr>
<td>Loose</td>
<td>87.7</td>
<td>88.1</td>
</tr>
</tbody>
</table>
Softmax-Margin beats CLL

Test set results

<table>
<thead>
<tr>
<th>C&amp;C '07</th>
<th>DecF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tight beam</td>
<td>87.7</td>
</tr>
<tr>
<td>Loose beam</td>
<td>87.7</td>
</tr>
</tbody>
</table>

+0.9
Softmax-Margin beats CLL

Does task-specific optimisation degrade accuracy on other metrics?
Does task-specific optimisation degrade accuracy on other metrics?

Softmax-Margin beats CLL
Integrated Model + SMM

Marcel proved completeness
Integrated Model + SMM

Marcel proved completeness
Integrated Model + SMM

forward-backward

Marcel

proved

completeness

Hamming augmented expectations
Integrated Model + SMM

- forward-backward
- Marcel
- proved
- completeness
- Hamming
- augmented
- expectations
Integrated Model + SMM

- parsing factor
- forward-backward
- supertagging factor
- Marcel
- proved
- completeness
- Hamming augmented expectations
Integrated Model + SMM

- Marcel
- proved
- completeness
- parsing factor
- inside-outside
- supertagging factor
- forward-backward
- F-measure
- augmented expectations
- Hamming
- augmented expectations
- completeness
Integrated Model + SMM

inside-outside

parsing factor

forward-backward

supertagging factor

Marcel

proved

completeness

F-measure augmented expectations

Hamming augmented expectations
Results: Integrated Model

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.
Results: Integrated Model

- F-measure loss for parsing sub-model (+DecF$_1$).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.

![Bar chart showing labelled F-measure results](chart.png)

- C&C ’07
- Integrated
- +DecF$_1$
- +Tagger
Results: Integrated Model

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.

![Bar chart showing labelled F-measure results](chart.png)

- C&C ’07
- Integrated
- +DecF₁
- +Tagger
Results: Integrated Model

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.
Results: Integrated Model

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.

C&C ’07
Integrated
+DecF₁
+Tagger
Results: Automatic POS

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.

Fowler & Penn (2010)
Results: Automatic POS

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.

Fowler & Penn (2010)
Results: Automatic POS

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.
Results: Automatic POS

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.

Fowler & Penn (2010)
Results: Automatic POS

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.
Results: Automatic POS

- F-measure loss for parsing sub-model (+DecF₁).
- Hamming loss for supertagging sub-model (+Tagger).
- Belief propagation for inference.

Fowler & Penn (2010)
Results: Efficiency vs. Accuracy

Faster

Sentences/second

Better

Accuracy
Results: Efficiency vs. Accuracy

Faster

Accumtry

Sentences/second

Better

0 11
0 99
0 88
0 77
0 66
0 55
0 44
0 33
0 22
0 11
0 0

87 88 89 90
Results: Efficiency vs. Accuracy

- **Faster**
- **Accuracy**

The graph shows a point labeled "C&C" indicating a trade-off between efficiency and accuracy, with faster values on the y-axis and accuracy on the x-axis.
Results: Efficiency vs. Accuracy

- Faster
- Better

Accuracy

Sentences/second

C&C
Results: Efficiency vs. Accuracy

- **Faster**: Sentences/second
- **Better**: Accuracy

Graph showing comparison between C&C and Integrated Model.
Results: Efficiency vs. Accuracy

![Graph showing efficiency vs. accuracy with points representing different models: Faster (Sentences/second) on the y-axis and Accuracy on the x-axis. Points labeled C&C, Integrated Model, Softmax-Margin Training.](image)
Summary

- Softmax-Margin training is easy and improves our model.

- Approximate loss functions are fast, accurate and easy to use.

- Best ever CCG parsing results (87.7 → 89.3).
Future Directions

• What can we do with the presented methods?
  • BP for other complex problems e.g. SMT
  • Semantics for SMT.
  • Simultaneous parsing of multiple sentences.
BP for other NLP pipelines

- Pipelines necessary for practical NLP systems
- More accurate integrated models often too complex
- This talk: Approximate inference can make these models practical
- Use it for other pipelines e.g. POS, NER tagging & Parsing
- Hard: BP for syntactic MT, another weighted intersection problem between LM & TM
Semantics for SMT

• Compositional & distributional meaning representation to compute vectors of sentence-meaning (Greffestette & Sadrzadeh, 2011; Clark, to appear)

• Syntax (e.g. CCG) drives compositional process

• Directions: Model optimisation, evaluation, LM
Many NLP tasks (e.g. IE) rely on uniform analysis of constituents.

Skip-Chain CRFs successful to predict consistent NER tags across sentences (Sutton & McCallum, 2004).

Parse multiple sentences at once and enforce uniformity of parses.

1. 
   [The securities and exchange commission](NP/N) issued ...
   \[NP/N\quad N\quad NP\]

2. 
   [... responded to the statement of the securities and exchange commission](NP/NP)
   \[NP\quad conj\quad NP\quad \text{NP}\backslash\text{NP}\quad NP\]
Related Publications


Thank you