Combining Dynamic Programming and Integer Linear Programming for Dependency Parsing

Alexis Nasr - Ghasem Mirroshandel

Laboratoire d’Informatique Fondamentale de Marseille
Motivations

- Statistical parsers are trained on few thousand sentences manually annotated with syntax
- Some syntactico-lexical phenomena are incorrectly modeled, due to treebanks size
  - Jean regarde un homme avec un télescope
  - manger une glace à la fraise
  - commander une glace à la fraise
  - commander une glace à la serveuse
  - système de communication rapide
- We will never have enough annotated data for such phenomena!
Semi-supervised learning

- But: we have a lot of raw text
- General Idea:
  - Train a parser on a treebank
  - Parse a large amount of raw text
  - Select interesting sub-parses
  - Integrate this new data in the parser
Outline

Parsing
  Dependency Structures
  Graph-based Parsing
  Results on French data

Patching
  Lexico-Syntactic Configurations
  Subcategorization Frames
  Selectional Constraints
  Combining Subcategorization Frames and Selectional Constraints

Combining Parsing and Patching

Experiments
  Subcat Frames
  Selectional Constraints
La diane chantait dans les cours des casernes
The trumpet blew in the yards of the barracks
### CONLL Format

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>La</td>
<td>le</td>
<td>ART</td>
<td>2</td>
<td>DET</td>
</tr>
<tr>
<td>2</td>
<td>diane</td>
<td>diane</td>
<td>N</td>
<td>3</td>
<td>SUJ</td>
</tr>
<tr>
<td>3</td>
<td>chantait</td>
<td>chanter</td>
<td>V</td>
<td>0</td>
<td>ROOT</td>
</tr>
<tr>
<td>4</td>
<td>dans</td>
<td>dans</td>
<td>PREP</td>
<td>3</td>
<td>MOD</td>
</tr>
<tr>
<td>5</td>
<td>les</td>
<td>le</td>
<td>ART</td>
<td>6</td>
<td>DET</td>
</tr>
<tr>
<td>6</td>
<td>cour</td>
<td>cour</td>
<td>N</td>
<td>4</td>
<td>OBJ</td>
</tr>
<tr>
<td>7</td>
<td>des</td>
<td>de</td>
<td>PREP</td>
<td>6</td>
<td>MOD</td>
</tr>
<tr>
<td>8</td>
<td>casernes</td>
<td>caserne</td>
<td>N</td>
<td>7</td>
<td>OBJ</td>
</tr>
</tbody>
</table>
Projectivity

- Property defined on ordered trees
- A dependent cannot be separated from its governor in the string by a word that is not a descendent of the governor

▶ Linguistically reasonable
▶ Important reduction of the search space

<table>
<thead>
<tr>
<th>nodes nb</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>trees</td>
<td>9</td>
<td>64</td>
<td>625</td>
<td>7776</td>
<td>117649</td>
<td>2.10^6</td>
<td>43.10^6</td>
<td>1.10^9</td>
</tr>
<tr>
<td>proj. trees</td>
<td>7</td>
<td>30</td>
<td>143</td>
<td>728</td>
<td>3876</td>
<td>21318</td>
<td>1200001</td>
<td>690690</td>
</tr>
<tr>
<td>ratio</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>11</td>
<td>30</td>
<td>98</td>
<td>358</td>
<td>1448</td>
</tr>
</tbody>
</table>
Graph-based parsing

- No rewriting grammar
- For a given sentence $S = m_1 \ldots m_n$ and a functional labels tagset $\mathcal{F}$ any dependency tree $T$ for $S$ is a possible syntactic analysis of $S$.
- One structural constraint : projectivity
- Score of a tree :
  \[ s(T) = \sum_{\psi \in \psi(T)} s(\psi) \]
  - $\psi(T)$ is the set of all relevant parts of $T$
  - $s(\psi)$ is the score of $\psi$
Decomposition

- First order models: a relevant part is just a dependency

- Second order models: a relevant part is one of

or
Example

chantait

SUJ

diane

MOD
dans

DET

La

cours

DET

les

MOD
casernes

DET

chantait
diane
dans

SUJ

La

des

cours
casernes

DET

les

cours

DET

des

cours

MOD

La

les

casernes

OBJ

DET
Score of a part

- A part is decomposed as a vector $f$ of elementary features.
  
  \[
  (\text{diane}, \text{diane}, \text{N}) \xrightarrow{\text{-DET}} (\text{la}, \text{le}, \text{ART})
  \]
  
  \[
  (X, X, \text{N}) \xrightarrow{\text{-DET}} (X, X, \text{ART})
  \]
  
  \[
  (\text{diane}, X, \text{N}) \xrightarrow{\text{-DET}} (X, X, \text{ART})
  \]
  
  \[
  (X, X, \text{N}) \xrightarrow{\text{-DET}} (\text{la}, X, \text{ART})
  \]
  
  \[
  (X, \text{diane}, \text{N}) \xrightarrow{\text{-DET}} (X, \text{le}, \text{ART})
  \]
  
  ...

- A score $w$ is computed for every feature.

- The score of the part is the dot product $f \cdot w$.

- Weights are computed with an on-line machine learning algorithm (perceptron, MIRA)
Decoding

\[ \hat{T} = \arg\max_{T \in \mathcal{T}(S)} \sum_{X \in \psi(T)} s(X) \]

\[ \mathcal{T}(S) \] is the set of all projective trees for sentence \( S \)

- Enumerate all projective trees for a sentence
- Compute the score of each tree
- Select the tree with highest score
- Dynamic Programming: \( O(n^3) \)
Constrained Parsing

- Force the parser to produce a solution that contains dependency $\delta = (g, r, d)$
- Define a new weight function $s^+_\delta$ based on $s$

$$s^+_\delta(g', r', d') = \begin{cases} 
-\infty & \text{if } d' = d \text{ and } (g' \neq g \text{ or } r' \neq r) \\
 s(g', r', d') & \text{otherwise}
\end{cases}$$
Constrained Parsing

Force the parser to produce a solution that contains dependency set $\Delta$

$$s^+_\Delta(g, r, d) = \begin{cases} 
  s^+_\delta(g, r, d) & \text{if } (\cdot, \cdot, d) \in \Delta \\
  s(g, r, d) & \text{otherwise}
\end{cases}$$
Simple Confidence Measure

- k best parses of a sentence
- subtree $\Delta$ present in at least one of the k best parses
- $C(\Delta)$: number of occurrences of $\Delta$ in the k trees
- $CM(\Delta)$: confidence measure associated to $\Delta$

$$CM(\Delta) = \frac{C(\Delta)}{k}$$
Results on French data

French Treebank

<table>
<thead>
<tr>
<th></th>
<th>sent. nb.</th>
<th>tokens nb.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAIN</td>
<td>9 881</td>
<td>278 083</td>
</tr>
<tr>
<td>DEV</td>
<td>1 239</td>
<td>36 508</td>
</tr>
<tr>
<td>TEST</td>
<td>1 235</td>
<td>36 340</td>
</tr>
</tbody>
</table>

Accuracy

<table>
<thead>
<tr>
<th></th>
<th>TEST</th>
<th>DEV</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAS</td>
<td>88.88</td>
<td>88.53</td>
</tr>
<tr>
<td>UAS</td>
<td>90.71</td>
<td>90.37</td>
</tr>
</tbody>
</table>

UAS  ratio of words with correct governor
LAS  ratio of words with correct governor and correct syntactic function
Errors type distribution

![Graph showing error ratio against error type]

- Error Type
  - Error ratio

- Error Type range: 0 to 100
- Error ratio range: 0 to 0.14
## Most frequent errors on DEV

<table>
<thead>
<tr>
<th>dependency</th>
<th>freq.</th>
<th>acc.</th>
<th>contrib.</th>
</tr>
</thead>
<tbody>
<tr>
<td>N→N</td>
<td>1.50</td>
<td>72.23</td>
<td>2.91</td>
</tr>
<tr>
<td>V → à</td>
<td>0.88</td>
<td>69.11</td>
<td>2.53</td>
</tr>
<tr>
<td>V—suj → N</td>
<td>3.43</td>
<td>93.03</td>
<td>2.53</td>
</tr>
<tr>
<td>N → CC</td>
<td>0.77</td>
<td>69.78</td>
<td>2.05</td>
</tr>
<tr>
<td>N → de</td>
<td>3.70</td>
<td>92.07</td>
<td>2.05</td>
</tr>
<tr>
<td>V → de</td>
<td>0.66</td>
<td>74.68</td>
<td>1.62</td>
</tr>
<tr>
<td>V—obj → N</td>
<td>2.74</td>
<td>90.43</td>
<td>1.60</td>
</tr>
<tr>
<td>V → en</td>
<td>0.66</td>
<td>81.20</td>
<td>1.24</td>
</tr>
<tr>
<td>V → pour</td>
<td>0.46</td>
<td>67.78</td>
<td>1.10</td>
</tr>
<tr>
<td>N → ADJ</td>
<td>6.18</td>
<td>96.60</td>
<td>0.96</td>
</tr>
<tr>
<td>N → à</td>
<td>0.29</td>
<td>70.64</td>
<td>0.72</td>
</tr>
<tr>
<td>N → pour</td>
<td>0.12</td>
<td>38.64</td>
<td>0.67</td>
</tr>
<tr>
<td>N → en</td>
<td>0.15</td>
<td>47.69</td>
<td>0.57</td>
</tr>
</tbody>
</table>
Lexico-Syntactic Configurations (LSC)

- Pair \((s, T)\)
  - \(s\) is a score
  - \(T\) is a dependency tree of arbitrary size

- Nodes of \(T\) are 5-tuples \((f, p, w, l, i)\)
  - \(f\) functional label
  - \(p\) part of speech tag
  - \(l\) lemma
  - \(w\) word
  - \(i\) index (position in a sentence)
Example

(0.028, [\textit{V:offrir::}](
  \text{[SUJ:N::]},
  \text{[OBJ:N:fleur:fleurs:]},
  \text{[AOBJ:PREP:a:a:]}(\text{[OBJ:N::]})))

- fields can be unspecified in LSC
- indices are always unspecified in LSC
Instantiated Lexico-Syntactic Configurations (ILSC)

- result of instantiating an LSC on a sentence
- preceding LSC instantiates on sentence

Jean offre des fleurs à Marie

as

(0.028, [:V:offrir:offre:2](
    [SUJ:N:Jean:Jean:1],
    [OBJ:N:fleur:fleurs:4],
    [AOBJ:PREP:a:a:5](
        [OBJ:N:Marie:Marie:6])))
Patching = Selecting the Optimal Set of ILSC

- Given a sentence $S$, and a set $L$ of LSC
- $\mathcal{I}$ is the set of all possible instantiations of elements of $L$ on $S$.
- $\hat{\mathcal{I}} \subseteq \mathcal{I}$ is the set of *compatible* ILSC such that

$$\hat{\mathcal{I}} = \arg \max_{\mathcal{I}' \subseteq \mathcal{I}} \sum_{x \in \mathcal{I}'} s(x)$$

- $s(x)$ is the score of ILSC $x$
- Cannot be done by brute force
- Integer Linear Programming
Subcat Frames (SF)

- Special kind of LSC
  - root = predicate
  - leaves = arguments

- Example
  
  \[(0.028, [:V:donner::] ( [SUJ:N:::], [OBJ:N:::], [AOBJ:PREP:a:a:] ( [OBJ:N:::] ))))\]

- score \(s_{SF} = P(T|\nu)\)
Selecting the Optimal Set of Subcategorization Frames

▶ **Notations**

- $R(j)$ predicate of ISF $j$
- $L(j)$ arguments of ISF $j$

▶ **Definition of the variables**

- $\alpha^i_j = 1$ if word $i$ is the predicate of ISF number $j$, 0 otherwise
- $\beta^i_j = 1$ if word $i$ is an argument of ISF number $j$, 0 otherwise

▶ **Definition of the constraints**

- a word is the predicate of at most one ISF:
  \[ \forall i \in \{1, \ldots, N\} \sum_{j \in I} \alpha^i_j \leq 1 \]
- a word cannot be an argument of more than one ISF:
  \[ \forall i \in \{1, \ldots, N\} \sum_{j \in I} \beta^i_j \leq 1 \]
- for an ISF to be selected, its pred. and all its args must be:
  \[ \forall j \in \{1, \ldots, |I|\} |L(j)|\alpha^{R(j)}_j - \sum_{l \in L(j)} \beta^l_j = 0 \]

▶ **Definition of the objective function**

\[
\max \sum_{j \in I} \alpha^j_{R(j)} S_{SF}(j)
\]
Example

\[ S = \text{Jean rend le livre qu’il a emprunté à la bibliothèque.} \]
\((\text{Jean returns the book that he has borrowed at the library.})\)

1 \((0.2, [:V:rend:rendre:2](\]
\[ [SUJ:N:Jean:Jean:1],\]
\[ [OBJ:N:livre:livre:4]))\)

2 \((0.4, [:V:rend:rendre:2](\]
\[ [SUJ:N:Jean:Jean:1],\]
\[ [OBJ:N:livre:livre:4],\]
\[ [AOBJ:PREP:a:a:9](\]
\[ [OBJ:N:biblio.:biblio.:11]))\))

3 \((0.3, [:V:emprunte:emprunter:8] (\]
\[ [SUJ:N:il:il:6],\]
\[ [OBJ:N:qu’:que:5])\))

4 \((0.6, [:V:emprunte:emprunter:8] (\]
\[ [SUJ:N:il:il:6],\]
\[ [OBJ:N:qu’:que:5],\]
\[ [OBJ:N:biblio.:biblio.:11]))\))

\[ \widehat{L} = \{1, 4\}. \]
Selectional Constraints (SC)

- Special kind of LSC
  - one lexical root
  - one lexical leaf

- tendency of the root and the leaf to co-occur in a specific syntactic configuration.

- four configurations:
  - \([\text{:V:::}][\text{SUJ:N:::}]\)
  - \([\text{:V:::}][\text{OBJ:N:::}]\)
  - \([\text{:V:::}][\text{DOBJ:P:de::OBJ:N:::}]\)
  - \([\text{:V:::}][\text{AOBJ:P:a::OBJ:N:::}]\)
SC Scores

- The score of a SC reflects the tendency of the root $r$ and the leaf $l$ to appear together in configuration $C$.

- It is maximal if:
  - whenever $r$ occurs as the root of configuration $C$, the leaf position is occupied by $l$
  - and, symmetrically, if whenever $l$ occurs as the leaf of configuration $C$, the root position is occupied by $r$.

\[
s_{SC}(C, r, l) = \frac{1}{2} \left( \frac{C(C, r, l)}{C(C, *, l)} + \frac{C(C, r, l)}{C(C, r, *)} \right)
\]

- $C(C, l, *)$ : occurrences of conf. $C$ with $l$ as a root
- $C(C, *, l)$ : occurrences of conf. $C$ with $l$ as a leaf
- $C(C, r, l)$ : occurrences of conf. $C$ with $r$ as a root and $l$ as a leaf.
Selecting the Optimal Set of Selectional Constraints

Definition of the variables

- \( \gamma^j_i = 1 \) if word \( i \) is the root of ISC number \( j \), 0 otherwise
- \( \delta^j_i = 1 \) if word \( i \) is the leaf of ISC number \( j \), 0 otherwise

Definition of the constraints

- A word cannot be the leaf of more than one ISC
  \[ \forall i \in \{1, \ldots, N\} \sum_{j \in \mathcal{I}'} \delta^j_i \leq 1 \]
- For ISC \( j \) to be selected, both its root and its leaf must be
  \[ \forall j \in \{1, \ldots, |\mathcal{I}'|\} \gamma^j_{R(j)} - \delta^j_{d \in \mathcal{L}(j)} = 0 \]

Definition of the objective function

\[ \max \sum_{j \in \mathcal{I}'} \gamma^j_{R(j)} \text{ssc}(j) \]
Example

\[ S = \text{Jean rend le livre qu’il a emprunté à la bibliothèque}. \]

5 (0.2, [:V:rend:rendre:2]
   ([SUJ:N:Jean:Jean:1]))
6 (0.2, [:V:rend:rendre:2]
   ([OBJ:N:livre:livre:4]))
7 (0.4, [:V:rend:rendre:2]
   ([AOBJ:PREP:a:a:9]
    ([OBJ:N:biblio.:biblio.:11])))
8 (0.6, [:V:emprunte:emprunter:8]
   ([AOBJ:PREP:a:a:9]
    ([OBJ:N:biblio.:biblio.:11])))

\[ \hat{I} = \{ 5, 6, 8 \}. \]
Combining Subcategorization Frames and Selectional Constraints

- **Definition of the variables**
  \[ \alpha_i^j, \beta_i^j, \gamma_i^j, \delta_i^j \]

- **Definition of the constraints**
  - All the constraints of ISF selection and ISC satisfaction
  - Incompatible ISF and ISC cannot be selected together
    An ISF \( j \) and an ISC \( j' \) are not compatible if they share a common leaf but have different roots.
    \[ \forall i, i', j, j' \alpha_i^j + 2\beta_i^j + \gamma_i^{j'} + 2\delta_i^{j'} \neq 5 \]

- **Definition of the objective function**
  \[ \max \left( \sum_{j \in I'} \delta_{R(j)}^j s_{SC}(j) + \sum_{j \in I} \alpha_{R(j)}^j s_{SF}(j) \right) \]
Two Processes

- **Dynamic Programming**: Parsing

\[
\hat{T} = \arg\max_{T \in T(S)} \sum_{X \in \Psi(T)} s(X)
\]

- **Linear Programming**: Patching

\[
\hat{C} = \arg\max_{E \subseteq C(S)} \sum_{X \in E} s'(X)
\]

- **How to combine them?**
Combining

- **Strong integration**
  - Parsing as an Integer Program
  - Patching as a Dynamic Programming problem

- **Weak integration**
  - Run the processes separately
  - Combine the outputs
Weak integration

1 Candidate generation
   ▶ Produce k best parses of sentence S.
   ▶ Build set $\mathcal{I}$ of ISFs and set $\mathcal{I}'$ of ISCs.

2 Candidate selection
   ▶ Patch $S$ with $\mathcal{I}$ and $\mathcal{I}'$ to get set $\mathcal{I}''$ of ISF and ISC.

3 Constrained Parsing
   ▶ Compute new scoring function $s_{\mathcal{I}''}^+$
   ▶ Produces a parse tree $\hat{T}$ that preserves the ISF and ISC computed in step 2.
Bias

- Patching always win!
- integrating confidence measure in the patching process:

\[ \hat{s}_{SF}(\Delta) = (1 - \mu_{SF})s_{SF}(\Delta) + \mu_{SF}CM(\Delta) \]

\[ \hat{s}_{SC}(\Delta) = (1 - \mu_{SC})s_{SC}(\Delta) + \mu_{SC}CM(\Delta) \]
## Raw Data

<table>
<thead>
<tr>
<th>CORPUS</th>
<th>Sent. nb.</th>
<th>Tokens nb.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFP</td>
<td>2 041 146</td>
<td>59 914 238</td>
</tr>
<tr>
<td>EST REP</td>
<td>2 998 261</td>
<td>53 913 288</td>
</tr>
<tr>
<td>WIKI</td>
<td>1 592 035</td>
<td>33 821 460</td>
</tr>
<tr>
<td>TOTAL</td>
<td>5 198 642</td>
<td>147 648 986</td>
</tr>
</tbody>
</table>
SF extraction

- Linguistic constraints
  - category of the root $\in \{V, \text{VINF}, \text{VPP}, \text{VPR}\}$
  - functional labels $\in \{\text{Suj}, \text{Obj}, \text{A_Obj}, \text{DE_Obj}\}$
  - category of the pre leaves $\in \{\text{P}, \text{CS}\}$
  - category of the leaves $\in \{\text{ADJ, N, V, VINF, VPP, VPR}\}$

- Abstraction
  - abstract over linear order
  - group together proper nouns, common nouns and pronouns into a single category N
Some statistics

<table>
<thead>
<tr>
<th></th>
<th>$A_0$</th>
<th>$A_5$</th>
<th>$A_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>nb of verbs</td>
<td>23,915</td>
<td>4,871</td>
<td>3,923</td>
</tr>
<tr>
<td>nb of diff SF</td>
<td>12,122</td>
<td>2,064</td>
<td>1,355</td>
</tr>
<tr>
<td>avg. nb of SF</td>
<td>14.26</td>
<td>16.16</td>
<td>13.45</td>
</tr>
</tbody>
</table>
## Coverage

<table>
<thead>
<tr>
<th></th>
<th>Lex. cov.</th>
<th>Synt. cov.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>types</td>
<td>types</td>
</tr>
<tr>
<td>$\Lambda_0$</td>
<td>99.52</td>
<td>95.78</td>
</tr>
<tr>
<td>$\Lambda_5$</td>
<td>98.56</td>
<td>91.08</td>
</tr>
<tr>
<td>$\Lambda_{10}$</td>
<td>98.08</td>
<td>88.84</td>
</tr>
<tr>
<td>$T$</td>
<td>89.56</td>
<td>62.24</td>
</tr>
<tr>
<td></td>
<td>tokens</td>
<td>tokens</td>
</tr>
<tr>
<td>$\Lambda_0$</td>
<td>99.85</td>
<td>97.13</td>
</tr>
<tr>
<td>$\Lambda_5$</td>
<td>99.62</td>
<td>93.96</td>
</tr>
<tr>
<td>$\Lambda_{10}$</td>
<td>99.50</td>
<td>92.39</td>
</tr>
<tr>
<td>$T$</td>
<td>96.98</td>
<td>73.54</td>
</tr>
</tbody>
</table>
Statistics

<table>
<thead>
<tr>
<th>Conf.</th>
<th>$A_0$</th>
<th>$A_5$</th>
<th>$A_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ</td>
<td>422 756</td>
<td>58 495</td>
<td>26 847</td>
</tr>
<tr>
<td>SBJ</td>
<td>433 196</td>
<td>55 768</td>
<td>25 291</td>
</tr>
<tr>
<td>VdeN</td>
<td>116 519</td>
<td>11 676</td>
<td>4 779</td>
</tr>
<tr>
<td>VaN</td>
<td>185 127</td>
<td>23 674</td>
<td>10 729</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1 157 598</td>
<td>149 613</td>
<td>67 646</td>
</tr>
</tbody>
</table>
## Coverage

<table>
<thead>
<tr>
<th>Conf.</th>
<th>types</th>
<th>(A_0)</th>
<th>(A_5)</th>
<th>(A_{10})</th>
<th>(T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>types</td>
<td>68.31</td>
<td>58.50</td>
<td>53.24</td>
<td>17.94</td>
<td></td>
</tr>
<tr>
<td>tokens</td>
<td>69.71</td>
<td>61.51</td>
<td>56.12</td>
<td>21.31</td>
<td></td>
</tr>
<tr>
<td>SBJ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>types</td>
<td>66.26</td>
<td>52.87</td>
<td>47.82</td>
<td>23.11</td>
<td></td>
</tr>
<tr>
<td>tokens</td>
<td>68.18</td>
<td>57.24</td>
<td>52.94</td>
<td>24.48</td>
<td></td>
</tr>
<tr>
<td>VdeN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>types</td>
<td>46.28</td>
<td>34.93</td>
<td>31.66</td>
<td>11.57</td>
<td></td>
</tr>
<tr>
<td>tokens</td>
<td>56.61</td>
<td>49.57</td>
<td>46.73</td>
<td>15.77</td>
<td></td>
</tr>
<tr>
<td>VaN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>types</td>
<td>62.93</td>
<td>48.89</td>
<td>42.68</td>
<td>20.06</td>
<td></td>
</tr>
<tr>
<td>tokens</td>
<td>64.69</td>
<td>52.42</td>
<td>47.56</td>
<td>23.34</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>types</td>
<td>65.17</td>
<td>53.10</td>
<td>47.92</td>
<td>19.96</td>
<td></td>
</tr>
<tr>
<td>tokens</td>
<td>67.54</td>
<td>57.15</td>
<td>52.78</td>
<td>22.24</td>
<td></td>
</tr>
</tbody>
</table>
Accuracy results

<table>
<thead>
<tr>
<th></th>
<th>LAS</th>
<th>UAS</th>
<th>SCAS</th>
<th>SFAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>88.88</td>
<td>90.71</td>
<td>87.81</td>
<td>80.84</td>
</tr>
<tr>
<td>SF</td>
<td>89.54</td>
<td>91.37</td>
<td>88.44</td>
<td>84.57</td>
</tr>
<tr>
<td>SC</td>
<td>89.73</td>
<td>91.60</td>
<td>92.32</td>
<td>82.14</td>
</tr>
<tr>
<td>SF + SC</td>
<td>89.85</td>
<td>91.72</td>
<td>93.08</td>
<td>85.26</td>
</tr>
</tbody>
</table>

- **Labeled Accuracy Score (LAS)**
  ratio of correct labeled dependencies in $\hat{T}$.

- **Unlabeled Accuracy Score (UAS)**
  ratio of correct unlabeled dependencies in $\hat{T}$.

- **Subcategorization Frame Accuracy Score (SFAS)**
  ratio of verbs in $\hat{T}$ that have been assigned their correct SF.

- **Selectional Constraint Accuracy Score (SCAS)**
  ratio of correct occurrences of SC patterns in $\hat{T}$.
Future Work

- Better candidate generation: use of parse forests
- Patching with Framenet Frames
- Patching with Discourse Patterns