

# Tutorial: An Efficient Algorithm for Easy-First Non-Directional Dependency Parsing – writing the code

Aviad Reich

2012-07-21 Sat

## Contents

<b>1</b>	<b>How to contribute to this document</b>	<b>2</b>
<b>2</b>	<b>An intro to dep parsing</b>	<b>3</b>
2.1	Formats of the data . . . . .	3
2.2	Example sentences . . . . .	3
2.3	Rendering of trees in text and graphically . . . . .	3
<b>3</b>	<b>Intro to dep parsing using transition based parsing</b>	<b>3</b>
3.1	How shift reduce works . . . . .	3
3.2	Turning a tree into a sequence of shift reduce transitions . . .	3
<b>4</b>	<b>Malt like parsing</b>	<b>3</b>
4.1	Training a classifier to learn which transition is best at each step . . . . .	3
4.2	Typical features used for malt . . . . .	3
<b>5</b>	<b>Evaluation methods for dep parsing</b>	<b>3</b>
<b>6</b>	<b>Evaluation of our malt parser</b>	<b>3</b>
<b>7</b>	<b>Easy First</b>	<b>3</b>
7.1	Read the paper . . . . .	3
7.2	Quiz on the paper . . . . .	4
7.2.1	By Sections . . . . .	4
7.2.2	Perceptron Classifier . . . . .	4
7.2.3	Cython . . . . .	4

## 1 How to contribute to this document

This document exists in XHTML, latex, text and org-mode versions. Editing the first three is straightforward, but will force all future contributors (if such will exist) to use the same version/format, since this will break the compatibility of the different formats. Do this only if you're sure the future contributors will be OK with it, or you know they will not exist (for example: you are now completing the document). If you have done so, be sure to either label the other versions as outdated/irrelevant, or simply delete them.

Another option is to edit the text file 'tutorial.org' in you favorite text editor. If you ignore lines 5-15, it's pretty much straightforward to understand the usage: The number of left-aligned '\*' before a line determine it's 'depth' in the document (more is deeper), and elements are automatically nested in the immediate higher level item.

if you want  $T_{\text{E}}X$ , or  $L^A T_{\text{E}}X$ , include its code inside adjacent dollar signs: `$[latex code here]$`. Just look at 'tutorial.org' for examples.

To use the other many wonderful, yet simple to use, features of org-mode, like auto-numbering of items, footnotes and others, it's recommended you read the orgmode website and the (relevant) docs, or the tutorial that comes with it as part of Emacs, or these tutorials.

**Exporting** To export to HTML, tex and text once you're done, open Emacs (you know where to get it), press Ctrl-x Ctrl-f and type the path to the file. Then, press Ctrl-c Ctrl-e h (for HTML), Ctrl-c Ctrl-e p (for pdf), Ctrl-c Ctrl-e l (for latex), Ctrl-c Ctrl-e u (for unicode text, a for ascii). Many other export format exist - you'll find it in the "Emacs-style popup" window.

**MAKE SURE YOU EXPORT IN ALL FORMATS ONCE YOU'RE DONE, SO COMPATIBILITY IS KEPT.**

As this is written (August 2012), I care about this document, and would be happy to extend my help if it's wanted. To email me use the first 3 letters of 'Aviad', followed by a dot ('.') and the first 3 of 'Reich'. Then mail me at: [\[what-you-got\]@gmail.com](mailto:[what-you-got]@gmail.com).

Additionally, there is a github repo: [https://github.com/lxmonk/NLP12-Easyfirst\\_tutorial](https://github.com/lxmonk/NLP12-Easyfirst_tutorial), that you can clone or fork. If you do, and you've created

a new one - change this address. Otherwise, let me know and I'll update.

## 2 An intro to dep parsing

### 2.1 Formats of the data

### 2.2 Example sentences

### 2.3 Rendering of trees in text and graphically

## 3 Intro to dep parsing using transition based parsing

### 3.1 How shift reduce works

### 3.2 Turning a tree into a sequence of shift reduce transitions

## 4 Malt like parsing

### 4.1 Training a classifier to learn which transition is best at each step

### 4.2 Typical features used for malt

## 5 Evaluation methods for dep parsing

## 6 Evaluation of our malt parser

## 7 Easy First

### 7.1 Read the paper

The article:

**Easy First Dependency Parsing of Modern Hebrew,**

Yoav Goldberg and Michael Elhadad,

*SPMRL 2010 (NAACL Workshop on Statistical Parsing of Morphologically-rich Languages)*

It can be obtained from Yoav Goldberg's BGU webpage, or at the acm website: <http://dl.acm.org/citation.cfm?id=1857999.1858114>.

## **7.2 Quiz on the paper**

### **7.2.1 By Sections**

- Introduction

1. From the article (pg. 1):

Current dependency parsers can be categorized into three families: local-and-greedy transition- based parsers (e.g., MALTPARSER (Nivre et al., 2006)), globally optimized graph-based parsers (e.g., MST P ARSER (McDonald et al., 2005)), and hybrid systems (e.g., (Sagae and Lavie, 2006b; Nivre and McDonald, 2008)), which combine the output of various parsers into a new and improved parse, and which are orthogonal to our approach.

### **7.2.2 Perceptron Classifier**

### **7.2.3 Cython**

## **8 Cython primer**