CS 187 - Dependency Parsing
Phase 4: Final Paper Draft

Lauren Urke, Nathaniel Herman, Henrik Sigstad, Ariel Camperi

Abstract

The abstract ...

1 Introduction

- Motivation
- What we did and why we think it is important
- Statement of key result

2 Statement of the problem

- Further discussion about why we care...

3 Description of the method

3.1 Description of data

To implement Dependency Parsing, we used the Penn Treebank Project Data. The treebank data contains sentences that have been parsed into a linguistic trees that also contain part-of-speech tagging. Parsed in this way, "Pieere Vinken, 61 years old, will join the board as a non executive director Nov. 29.", will look like this:

NEED TO FIGURE OUT HOW TO TAB THIS

((S
(NP-SBJ
(NP (NNP Pierre) (NNP Vinken))
(, ,)
(ADJP
(NP (CD 61) (NNS years))
(JJ old))
(, ,))

(VP (MD will)
(VP (VB join)
(NP (DT the) (NN board))
(PP-CLR (IN as)
(NP (DT a) (JJ nonexecutive) (NN director)))
(NP-TMP (NNP Nov.) (CD 29))))
(. .)))

This information is helpful, but could be improved. To conduct analysis on the dependencies, we wanted to be able to easily infer the relationship between two words. Therefore, we used a program from the Lund University Computer Science Department that converted the Penn Treebank parses into dependency parses. For each word in a sentence, a dependency parse indicates the parent word and the part of speech. For example, the same sentence now comes out as this:

ID	Token	Part of Speech	Parent ID	Class
1	Pierre	NNP	2	NAME
2	Vinken	NNP	8	SBJ
3	,	,	2	P
4	61	CD	5	NMOD
5	years	NNS	6	AMOD
6	old	JJ	2	APPO
7	,	,	2	P
8	will	MD	0	ROOT
9	join	VB	8	VC
10	the	DT	11	NMOD
11	board	NN	9	OBJ
12	as	IN	9	ADV
13	a	DT	15	NMOD
14	nonexecutive	JJ	15	NMOD
15	director	NN	12	PMOD
16	Nov.	NNP	9	TMP
17	29	CD	16	NMOD
18	.	.	8	P

This data is now much more helpful for determining whether there is a direct parental relationship between two words in a sentence.

3.2 Description of SVMs

3.3 Description of dependency parsing

3.4 Description of code

Our actual implementation uses the same basic parsing algorithm as that in [**?**]

4 Results

4.1 Results

Dependency Accuracy	Root Accuracy	Complete Rate
0.89135	0.93675	0.32711

Dependency Accuracy	Root Accuracy	Complete Rate
0.88680	0.93894	0.32285

	(2,2)	(2,3)	(2,4)	(2,5)	(3,2)	(3,3)	(3,4)	(3,5)
Dep. Acc.	0.891	0.890	0.890	0.889	0.887	0.887	0.887	0.887
Root Acc.	0.937	0.928	0.934	0.930	0.932	0.929	0.928	0.927
Comp. Rate	0.327	0.330	0.325	0.326	0.312	0.318	0.316	0.317

5 Conclusion

- Summarize how well we were able to replicate the paper
- Ideas for improving their method?

References

[1] Yamada, H., and Matsumoto, Y. Statistical Dependency Analysis With Support Vector Machines. In *Proceedings of IWPT*, 2003.