Clause Identification

Antony R. Neu
neua@informatik.uni-freiburg.de

Efficient Natural Language Processing
Lehrstuhl für Algorithmen und Datenstrukturen
Lehrstuhlinhaberin: Prof. Hannah Bast
Universität Freiburg

Motivation

- POS Tagging
  - Words have a Part-Of-Speech tag
- Text chunking
  - Which words belong together
  - Not embedded or recursive
- Clause identification:
  - E.g. relative clauses
  - Recursive problem
  - Applications: text to speech

Overview

- Introduction
  - Problem definition
  - Applications
- Solutions
  - Rule-based approaches
  - Machine-Learning-based approaches
    - Demo
  - Hybrid systems
- Summary

Introduction - Definition

- Clause: *Group of words containing a subject and a predicate. Subject may be implicite.*
- Latin: claudere: close, conclude, enclose
- Two types:
  - Independent clause: sentence
  - Dependent clause:
    - sentence-like structure within a sentence
    - cannot exist without a main clause
- Examples:

1. *"The man, who is walking over the street, is my father." (DC/IC)*
2. *"He went to school and she went to work."* (IC/IC)

Introduction - Definition

- clause vs. phrase: phrase has no subject **and** predicate
- Examples:
  - a known writer
  - an entirely new culture
  - when they learn how to solve their problems with wikis
- Debatable definitions

Task to solve

- Clause identification (also: clause splitting, clause boundary recognition)
- Shared Task of CoNLL-2001 (Computational Natural Language Learning)
  - Find start and ending point of a clause
  - Determine clause structure of the sentence
  - Type of clause, e.g. relative clause, temporal clause is ignored
- Examples:
  - ((The space shuttle Atlantis blasted into orbit from Cape Canaveral) and (its crew launched the Galileo space probe on a flight to the planet Jupiter).)
  - (The deregulation of railroads and trucking companies (that (began in 1980)) enabled (shippers to bargain for transportation).)

Applications

- Text-To-Speech systems
- Machine-Translation
- Question-Answering
- Preprocessing for bilingual alignment
- Brokkoli?

CI vs. text chunking

”You will start to see shows where viewers program the program.”

- Chunked:

  (NP *You*) (VP will start to see) (NP shows) (ADVP where) (NP viewers) (VP program) (NP the program)

- Clauses:

  (S You will start to see shows (S where (S viewers program the program)) .)

- Nevertheless:
  - Fuzzy transitions
  - Some chunkers provide simple clause identification

CI vs. full parsing

- Clause identification as intermediate step (Ejerhed '90)
- Form of shallow parsing
- Full parsing: better precision
- Why not extract clauses from full parse?
  - Classification frameworks:
    - Faster (e.g. needed for question answering)
    - Easier to implement
    - More easily portable to new languages

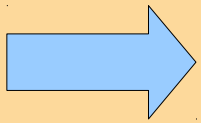
Solutions and Implementations

- Rule-Based-Systems (1990s)
- Machine Learning based systems (2000s)
- Hybrid systems (late 2000s)

Rule-based systems

- Clauses identified by predefined rules
- POS tags and/or chunk tags are taken into consideration
- Disadavantages:
  - Human work needed
  - Not easily adaptable to other languages
- Example:

-1: <VP>
 0: <NP>
 1: ,
 2: say (past o. Present)
 3: <NP>



Mark 0 as end of clause boundary.

Rule-based systems

- Ejerhed '96:
  - Only independent clauses identified
  - Starts and end identified
    - *(There was something true in that) (what he said).*
  - Regular expressions and stochastic approach

    DL_MAD XX => DL_MAD <c> XX
    - DL_MAD: major delimiter (., ?, !)
- Papageorgiou '97
  - Adresses embedded clauses
  - Inspired by Abbney's Cascaded Analysis of Syntactic Structure (CASS) parser ('91) (Full parser)
  - Text is tokenized and tagged (Brill tagger)
  - Clause tag marking module
    - What marks the clause, e.g. ”if” or ”as if”
  - Partial parsing generates clause structure

Rule-based systems

- Leffa '98:
  - Considers POS tags and **valence** of verb
  - Valence: How many other words does the verb bind?
    - 0: (It) is raining. (not a real subject)
    - 1: The dog runs. (a subject)
    - 2: I hate maths. (a subject and an object)
  - Read sentence left to right and mark clause initiators/terminators.
  - Clauses are segmented and processed
  - Valence is considered
    - (I know (when I have time).)
    - (I work (when (I have time)).)

Evaluation

AUTHOR		*Precision*	*Recall*	$F_{\beta=1}$
Ejerhed (1988)	*Regular expression*	87.01	98.89	92.57
	Stochastic	95.07	96.01	95.54
Ejerhed (1996)	*SUC corpus*	100.00	95.80	97.85
	DI93 corpus	98.80	90.70	94.58
Papageorgiou (1997)		95.44	93.06	94.23
Leffa (1998)		–	95.00	–
Sang & Déjean (2001)	*Best CoNLL-2001*	84.82	73.28	78.63
Sang & Déjean (2001)	*CoNLL-2001 Baseline*	98.44	31.48	47.71

source: Master Thesis, Benjamin Hachey, University Edinburgh

- Not identical corpora used for evaluation
- No standard
- Interpretation: good results

Machine Learning Systems

- Used for CoNLL-2001 shared task
  - Baseline: Assign Clause start and end at start and end of each **sentence**
- Basic idea:
  - Systems learn on a specific training set.
  - Classification problem (see text chunking)
  - Features are considered, e.g. the last 3 words (POS and chunk tags)
  - Decision: Is this word the beginning of a clause?

Implementations

- Carreras and Marquez (shown today)
  - Boosted decision trees
  - Perceptrons (neural networks)
  - Both concepts outperform all other participiants
- Others:
  - Short-Term Memory based
  - Conditional Random Fields
  - Hidden Markov Model

Benchmark Results CoNLL 2001

test cor	precision	recall	F
[CM03]	87.99%	81.01%	84.36
[CMPR02]	90.18%	78.11%	83.71
[CM01]	84.82%	78.85%	81.73
[MP01]	70.85%	70.51%	70.68
[TKS01]	76.91%	65.22%	70.58
[PG01]	73.75%	64.56%	68.85
[Dej01]	72.56%	58.69%	64.89
[Ham01]	55.81%	49.49%	52.46
baseline	98.44%	33.88%	50.41

Results of CoNLL-2001 shared task

Carreras & Marquez systems

- [CM01]
  - Learning algorithm (modified Adaboost) is given large number of binary simple features
- 4 feature types are used:
  - Word window: Surrounding sequence of words with their POS tags
  - Chunk window: Surrounding chunk tags of a word
  - Sentence patterns from word a to b:
    - All occurences of punctuation marks, relative pronouns, conjunctions, the word ”that” with its POS tag and VP chunks between a and b

Carreras & Marquez systems

- Sentence features:
  - Number of occurences VP, WP (pronoun), WP$, punctuation mark, beginning/end of clauses, the word ”that” to the **left** and **right** hand side of the word
- Window size was tuned to 3

- Filtering-Ranking Perceptron Learning for Partial Parsing (2005)
  - Similar Features to CM'01
  - Perceptrons are used instead of Adaboost
  - Implementation: Phreco

Phreco - Demo

- Uses perceptrons to recognize chunks or clauses
- Carreras' dissertation
- A demo is shown
- File with 11 sentences

Phreco - Evaluation

- Run times (45 000 words):
  - Test data set A: 44min 33s (743 KB, 2012 sentences, 1.3s per sentence)
  - Test data set B: 39min 33s (623 KB, 1671 sentences, 1.4s per sentence)
- Over 1 second per sentence
- Excluding tagging and chunking time

Phreco - Profiling

%Time	ExclSec	CumulS	#Calls	sec/call	Csec/c	Name
98.4	2696.	2696.5	75439	0.0357	0.0357	ml::vperceptron_classify
0.54	14.68	14.685	133684	0.0000	0.0000	PHRECO::phrase_set::top_phrases
0.26	7.057	7.057	75439	0.0001	0.0001	mapping::map_features
0.15	4.048	4.681	89564	0.0000	0.0001	PHRECO::clausefex::word_window_features
0.12	3.284	813.738	2012	0.0016	0.4044	PHRECO::frclauser::optimal_hierarchy
0.11	3.105	3.105	136515	0.0000	0.0000	PHRECO::clausefex::sentence_counts
0.11	3.102	5.668	122628	0.0000	0.0000	PHRECO::clausefex::sentence_counts_features
0.09	2.586	2.586	89564	0.0000	0.0000	PHRECO::clausefex::chunk_window_features
0.09	2.434	798.19	699017	0.0000	0.0011	PHRECO::frclauser::predict_phrase
0.08	2.176	1908.734	61314	0.0000	0.0311	PHRECO::sefilter::predict_word_signal
0.06	1.614	1.614	183942	0.0000	0.0000	PHRECO::clausefex::signal_window_features
0.05	1.450	1.450	3	0.4833	0.4833	ml::vperceptron_read_from_file
0.05	1.424	1928.413	2012	0.0007	0.9585	PHRECO::sefilter::start_end_filtering
0.04	1.026	2743.091	2012	0.0005	1.3634	PHRECO::frclauser::process_sentence
0.03	0.827	0.827	75439	0.0000	0.0000	ml::new_fvinput

Pearl profile

Hybrid Systems

- Recent works based on previous ML and rule based works
- Basic idea:
  - Use machine learning approach
  - Resolve errors with rules
- Papers:
  - Sundar et. al. '08 (best values)
  - Also: Nguyen'07

Sundar et al 2008

- Uses Conditional random fields as ML approach
- Features used (word windows of 5):
  - Word itself
  - POS tag
  - Chunk tag
  - Can linguistic rules be applied? (used later)

Sundar et al 2008

- Error analyzer and linguistic rules:
  - Find wrongly marked clause boundaries
  - 'Error patterns' are used for identification, e.g. unbalanced starts and endings of clauses
  - Linguistic rules are applied to correct errors (inside out)
  - Example rule:

-1: <VP>
 0: <NP>
 1: <VP infinitive>

Mark position 0 as clause boundary start.

Sundar et al 2008 - Benchmark

S.No	System	Precision (%)	Recall (%)	Fmeasure (%)
1	CRFs	83.68%	78.65%	81.08%
2	**CRFs with linguistic Rules**	**92.06%**	**87.89%**	**89.04%**

S. No	References	Techniques	Precision	Recall	F1 mesaure
1	**Our method**	**CRFs + linguistic Rules**	**92.06%**	**87.89%**	**89.04%**
2	Carreras et al. 05	FR-Perceptron	88.17%	82.10%	85.03%
3	Vinh Van Nyugenet al 07	CRFs	**90.01%**	**78.98%**	**84.09%**
4	Carreras et al. 02	AdaBoost class	90.18%	78.11%	83.71%
5	Carreras et al. 01	AdaBoost class	84.82%	78.85%	81.73%
6	Monila and Pla 01	HMM	70.85%	70.51%	70.68%

Summary

- Time-expensive intermediate task
- Not a lot of open-source implementations available
  - Lots of POS taggers and chunkers
  - Lots of Full parsers , role labelers etc.
  - Missing: intermediate task
- Hybrid systems seem to be an interesting approach

Sources

- **Overviews**:
- **Recognising Clauses Using Symbolic and Machine Learning Approaches**,Master Thesis, Benjamin Hachey, University of Edinburgh, http://benhachey.info/pubs/diss_msc.pdf
- **Introduction to the CoNLL-2001 Shared Task: Clause Identification,** Tjon, Sang, Déjan, 2001, http://www.cnts.ua.ac.be/conll2001/clauses/
- **References**:
- [CM01]: Xavier Carreras and Luís Màrquez, **Boosting Trees for Clause Splitting.** In: Proceedings of CoNLL-2001, Toulouse, France, 2001. http://www.cnts.ua.ac.be/conll2001/pdf/07375car.pdf
- [TKS01] Erik F. Tjong Kim Sang, **Memory-Based Clause Identification**. In: Proceedings of CoNLL-2001, Toulouse, France, 2001.
- [Eje96] Eva Ejerhed, **Finite State Segmentation of Discourse into Clauses**. In "Proceedings of the ECAI '96 Workshop on Extended finite state models of language", ECAI '96, Budapest, Hungary, 1996.
- [Lef98] Vilson J. Leffa, **Clause processing in complex sentences.** In: "Proceedings of LREC'98", Granada, Espanha, 1998
- Papageorgiou, H. (1997), **Clause recognition in the framework of alignment**, in 'Proceedings of the 2nd Conference on Recent Advances in Natural Language Processing (RANLP-97)', Tzigov Chark, Bulgaria, pp. 417–425.

Sources

- **References**:
- **[MP01]** Antonio Molina and Ferran Pla, **Clause Detection using HMM**. In: Proceedings of CoNLL-2001, Toulouse, France, 2001.
- **Clause Boundary Identification Using Conditional Random Fields,** R. Vijay Sundar Ram and Sobha Lalitha Devi, AU-KBC Research Centre, 2008 Springer Berlin / Heidelberg
- **Filtering-Ranking Perceptron Learning for Partial Parsing**, Xavier Carreras, Lluís Màrquez and Jorge Castro, Machine Learning Journal, Special Issue on Learning in Speech and Language Technologies, Volume 60, Issue 1-3, pages 41-71, Sept. 2005