Multiword Expression Identification and Statistical Dependency Parsing

Mathieu Constant
Mathieu.Constant@univ-lorraine.fr

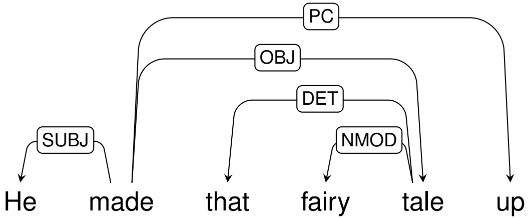
Université de Lorraine
ATILF, CNRS

Joint work with Marie Candito (Univ. Paris Diderot), Joseph Le Roux (Univ. Paris Nord), Joakim Nivre (Uppsala University) and Nadi Tomeh (Univ. Paris Nord)

Our task

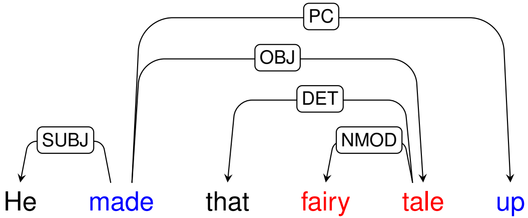
He made that fairy tale up

Our task



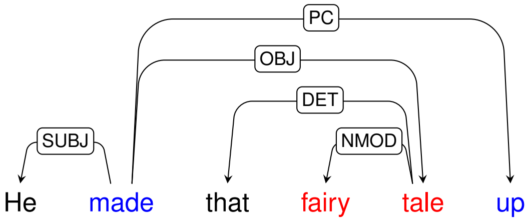
- syntactic analysis (dependency paradigm)

Our task



- syntactic analysis (dependency paradigm)
- lexical segmentation (multiword expressions)

Our task



- syntactic analysis (dependency paradigm)
- lexical segmentation (multiword expressions)

MWE-aware parsing

This talk

- Background: MWE processing, main approaches to MWE-aware parsing
- A transition-based system for joint lexical and syntactic analysis
  (Constant and Nivre ACL 2016)

Multiword Expressions (MWEs)

Definitional features

- A sequence of multiple lexemes that displays a certain degree of non-compositionality
- i.e. irregularity on one or more linguistic dimensions: morphological, lexical, syntactic, semantic, and pragmatic

Examples

- Nominal compounds: *grand-mère*, *cordon bleu*
- Adverbial compounds: *à fond*, *en dépit (de)*
- Grammatical compounds: *bien que*, *de la*
- Verbal idiomatic expressions: *casser les pieds*
- Light verb constructions: *prendre une décision*

MWE challenges for NLP I

Ambiguity

- MWE vs. literal meaning

  prendre la porte = *sortir* vs. *emporter la porte*

- MWE vs. accidental co-occurrence

  J'aime bien que tu viennes chez moi ***bien que*** *tu me fasses faire des bêtises*

Discontiguity

- *Luc* ***fait*** *souvent* ***face*** *à ce problème*
- *Luc* ***prend*** *cet argument* ***en compte***

MWE challenges for NLP II

Non-compositionality

- Various degrees of compositionality

  cordon bleu < eau de vie < arme blanche < appel d'offre
  casser les pieds < nager dans le bonheur < trembler de peur

- Internal compositional modifications

  prendre *une grande* ***décision***

Variability

- *Luc a cassé sa pipe/Luc et Marie ont cassé leur pipe*
- *Luc prend une décision/La décision prise par Luc me semble la bonne*

Embeddings

Luc (fait un (faux pas))

MWE Processing

MWE discovery

- **Task**: given a raw corpus, extract an MWE lexicon
- **Approaches**: linguistic patterns, association measures, modeling of MWE linguistic properties, distributional semantics, ...

MWE identification

- **Task**: given an input text and MWE resources, annotate occurrences of MWEs
- **Approaches**: rule-based identification based on lexicons, word sense disambiguation, supervised sequential tagging, ...

MWE Processing

MWE discovery

- **Task**: given a raw corpus, extract an MWE lexicon
- **Approaches**: linguistic patterns, association measures, modeling of MWE linguistic properties, distributional semantics, ...

MWE identification

- **Task**: given an input text and MWE resources, annotate occurrences of MWEs
- **Approaches**: rule-based identification based on lexicons, word sense disambiguation, supervised sequential tagging, ...

Motivations for MWE-aware parsing

MWE identification can help parsing

- MWEs constitute syntactic constituents
- Their identification can help syntactic attachments
  - rule-based parsing (Werhli et al. 2010, 2014)
  - statistical parsing (Cafferkey et al. 2007)

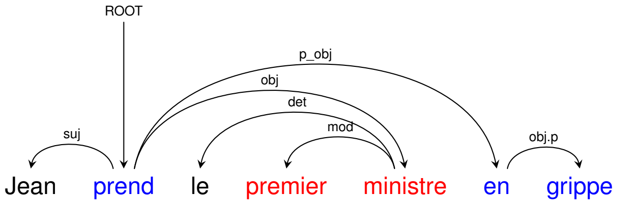
Parsing can help MWE identification

- help distinguish MWEs and accidental co-occurrences of words

  ex. French grammatical compounds (Nasr et al. 2015)
- help handle discontiguity and variability (Werlhi et al. 2010)

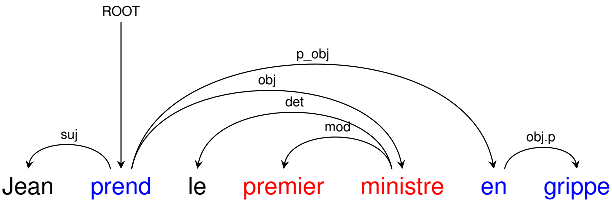
Our framework

- **Syntactic parsing** in combination with **MWE identification**
- **Supervised statistical approach:**
  - Training phase: annotated dataset → model
  - Annotation phrase: new raw data + model → annotated data
- **Use of lexical resources**
- **Example**



Our framework

- **Syntactic parsing** in combination with **MWE identification**
- **Supervised statistical approach:**
  - Training phase: annotated dataset → model (no details in this talk)
  - Annotation phrase: new raw data + model → annotated data
- **Use of lexical resources** (no details in this talk)
- **Example**



Few words on statistical dependency parsing

- No underlying grammatical formalism
- Parsing algorithms vary from local search (Nivre 2003) to global search (McDonald 2005)
- Discriminative approach
- Use of machine learning techniques: the deep learning revolution (Chen and Manning 2014, Dyer 2015, Weiss et al. 2015, Kiperwasser and Goldberg 2016)

Few words on statistical dependency parsing

- No underlying grammatical formalism
- Parsing algorithms vary from local search (Nivre 2003) to global search (McDonald 2005)
- Discriminative approach
- Use of machine learning techniques: the deep learning revolution (Chen and Manning 2014, Dyer 2015, Weiss et al. 2015, Kiperwasser and Goldberg 2016)

Statistical MWE-aware parsing Orchestration I

Three positions for MWE identification

- **Before parsing**: retokenization (*carte verte* → *carte_verte*)
  - gold (Nivre 2004, Arun 2005)
  - predicted (Cafferkey 2007, Constant et al. ACL 2012)
- **During parsing**: joint approach
  - Standard parsers (Nivre et Nilsson 2004, Arun and Keller 2005, ...)
  - Multilayer parsers (Constant et al. NAACL 2016, Constant and Nivre 2016)
- **After parsing**: Performing MWE identification on parsed text (Fazly et al. 2009)

→ Performances depend on the MWE types (Eryigit et al. 2011, Vincze et al. 2013)

Statistical MWE-aware parsing Orchestration II

Combining positions

- **before+during**:
  - *n*-best MWE tagger: lattice (Constant et al. ACM TSLP 2013) or beam (Urieli 2013) given to parser
  - dual decomposition: agreement on MWE segmentation for MWE taggers and joint parsers (Le Roux et al. COLING 2014)
  - reparser (Constant et al. SPMRL ST 2013)
- **during+after**:
  - *n*-best joint parser + MWE-based reranker (Constant et al. ACL 2012)

Joint approach using standard dependency parsers

Principle

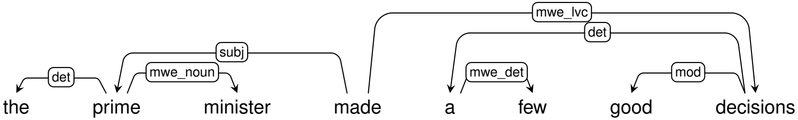
- **Each MWE is annotated as a subtree** of the syntactic tree in the reference treebank
- Use of **off-the-shelf parsers** that are learned from the reference treebank

How to represent MWEs ?

- **flat subtree** (Nivre and Nilsson 2004, Seddah et al. 2013, Nivre et al. 2016)
- **deep subtree** (Vincze et al. 2013, Candito and Constant ACL 2014)

Flat MWE representation

- MWE is annotated with a flat subtree within the syntactic tree
- The left-most (or right-most) MWE item is the head and other items are the modifiers
- Use of specific arc labels for MWE arcs



A dual MWE representation I

(Candito and Constant ACL 2014)

Irregular MWEs

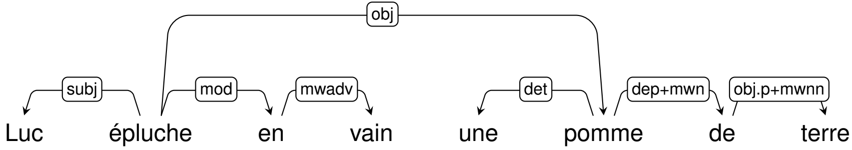
- They display irregular syntactic structure (e.g., *en vain* = Prep Adj)
- Use of flat MWE representation

Regular MWEs

- Internal syntactic structure is kept: use of classical syntactic dependency structure
- Arc label = syntactic label + MWE status

A dual MWE representation II

(Candito and Constant ACL 2014)



Multilayer lexical and syntactic parsing

Drawback of standard parsers

- No lexical embedding in dual representation
- |Label tagset| $\leq$ |MWE info| x |syntactic functions|
- Same mechanisms to predict lexical segmentation and syntactic structure

Principle

- Representations with two layers (or dimensions): lexical layer and syntactic layer
- Mild extension of dependency parsing algorithms

A Transition-based System for Joint Lexical and Syntactic Analysis

Contributions

A new factorized representation of lexical and syntactic analysis

- Dependency analysis
- Inclusion of Multiword Expression analysis

A new transition-based system

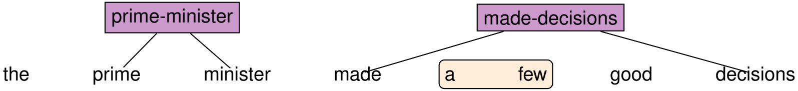
- Input: a sequence of tokens
- Output: above representation
- Special mechanisms to handle Multiword Expressions

Work originally presented at ACL 2016

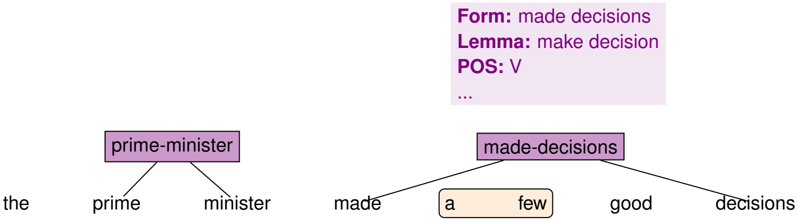
Lexical and Syntactic representation

the prime minister made a few good decisions

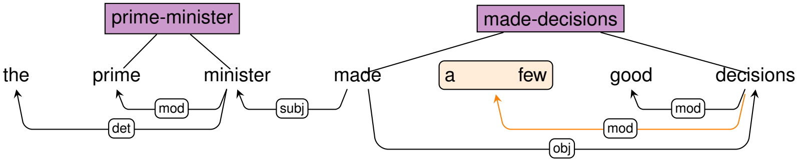
Lexical and Syntactic representation



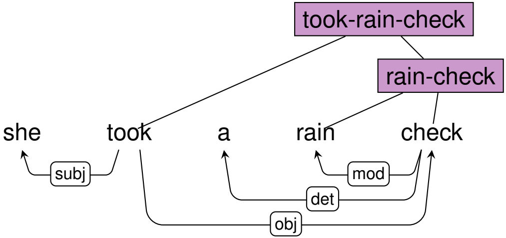
Lexical and Syntactic representation



Lexical and Syntactic representation



MWE embedding



Background: a standard transition-based parser

Input/Output

- **Input**: a sequence of tokens
- **Output**: a set of syntactic arcs

Internal mechanism

- predict a **sequence of actions** (namely *transitions*)
- A transition goes from one parsing state (namely *configuration*) to another one
- **Configuration**: a stack, a buffer and a set of arcs

Background: a standard transition-based parser (Cont'd)

Configurations

- **Initial configuration**: buffer filled with input tokens, empty stack and set of arcs
- **Terminal configuration**: buffer is empty, stack has one item left

Transitions

- **Shift**: push the next token of the buffer on top of stack
- **Left-arc**$_k$: creates a left arc labeled k between the two top tokens of the stack; only head item is kept in stack. The created arc is added to the set of arcs
- **Right-Arc**$_k$: same as Left-arc, but creates a right arc

Example

John likes linguistics

Transition

–

Buffer

[John likes linguistics]

Stack

[ ]

Arcs

–

Example

John likes linguistics

Transition
Shift

Buffer
[likes linguistics]

Stack
[John]

Arcs
–

Example

John likes linguistics

Transition
Shift

Buffer
[linguistics]

Stack
[John likes]

Arcs
–

Example

John likes linguistics

Transition
Left-Arc(subj)

Buffer
[linguistics]

Stack
[likes]

Arcs
subj(likes, John)

Example

John likes linguistics

Transition
Shift

Buffer
[ ]

Stack
[likes linguistics]

Arcs
subj(likes, John)

Example

John likes linguistics

Transition

Right-Arc(obj)

Buffer

[ ]

Stack

[likes]

Arcs

subj(likes, John)

obj(likes,linguistics)

Our new transition-based system

Handling two linguistic dimensions

- Two stacks: a syntactic stack and a lexical stack
- One buffer to synchronize the two dimensions
- Processed items: a set of syntactic arcs and a set of lexical trees

Handling MWEs

- Mild extension of arc-standard parser
- Specific transitions to deal with MWE identification

Transition system

Configuration

(Buffer, SynStack, SynArcs, LexStack, LexTrees)

Initial

$([w_1, \ldots, w_n], [\,], \{\,\}, [\,], \{\,\})$

Input: $w_1, \ldots, w_n$

Terminal

$([\,], [x], \text{SynArcs}, [\,], \text{LexTrees})$

Output: SynArcs, LexTrees

Transition system

Shift

Moves next token from Buffer to both stacks

Right-Arc(k), Left-Arc(k)

Adds syntactic arc between top items on syntactic stack

$\text{Merge}_F(t)$

Creates lexical tree from top items on both stacks – fixed MWE

$\text{Merge}_N(t)$

Creates lexical tree from top items on lexical stack – non-fixed MWE

Complete

Adds lexical tree from lexical stack

Example parse

Transition

–

Buffer

[he made a few decisions]

SynStack

[ ]

SynArcs

–

LexStack

[ ]

LexTrees

–

Example parse

Transition
Shift

Buffer
[made a few decisions]

SynStack
[he]

SynArcs
–

LexStack
[he]

LexTrees
–

Example parse

Transition
Complete

Buffer
[made a few decisions]

SynStack
[he]

SynArcs
–

LexStack
[ ]

LexTrees
he

Example parse

Transition
Shift

Buffer
[a few decisions]

SynStack
[he made]

SynArcs
–

LexStack
[made]

LexTrees
he

Example parse

Transition
Left-Arc(subj)

Buffer
[a few decisions]

SynStack
[made]

SynArcs
subj(made, he)

LexStack
[made]

LexTrees
he

Example parse

Transition
Shift

Buffer
[few decisions]

SynStack
[made a]

SynArcs
subj(made, he)

LexStack
[made a]

LexTrees
he

Example parse

Transition
Shift

Buffer
[decisions]

SynStack
[made a few]

SynArcs
subj(made, he)

LexStack
[made a few]

LexTrees
he

Example parse

Transition
$\text{Merge}_F(\text{A})$

Buffer
[decisions]

SynStack
[made A(a, few)]

SynArcs
subj(made, he)

LexStack
[made A(a, few)]

LexTrees
he

Example parse

Transition
Complete

Buffer
[decisions]

SynStack
[made A(a, few)]

SynArcs
subj(made, he)

LexStack
[made]

LexTrees
he, A(a, few)

Example parse

Transition
Shift

Buffer
[ ]

SynStack
[made A(a, few) decisions]

SynArcs
subj(made, he)

LexStack
[made decisions]

LexTrees
he, A(a, few)

Example parse

Transition
Left-Arc(mod)

Buffer
[ ]

SynStack
[made decisions]

SynArcs
subj(made, he)
mod(decisions, A(a, few))

LexStack
[made decisions]

LexTrees
he, A(a, few)

Example parse

Transition

$\text{Merge}_N(\text{V})$

Buffer

[ ]

SynStack

[made decisions]

SynArcs

subj(made, he)

mod(decisions, A(a, few))

LexStack

[V(made, decisions)]

LexTrees

he, A(a, few)

Example parse

Transition
Complete

Buffer
[ ]

SynStack
[made decisions]

SynArcs
subj(made, he)
mod(decisions, A(a, few))

LexStack
[ ]

LexTrees
he, A(a, few), V(made, decisions)

Example parse

Transition
Right-Arc(obj)

Buffer
[ ]

SynStack
[made]

SynArcs
subj(made, he)
mod(decisions, A(a, few))
obj(made, decisions)

LexStack
[ ]

LexTrees
he, A(a, few), V(made, decisions)

Implementation and Evaluation

Implementation

- **Greedy parser** trained with averaged **perceptron**
- **Hard constraints**: Complete transitions are made implicit, i.e. only activated when arc transitions are selected by classifier

Evaluation

- **Two datasets**: English Web Treebank (+ Streusle) and French Treebank
- **Comparisons** with
  1. standard parser with extended labels including the MWE status
  2. partial systems where some transitions are deactivated
  3. pipeline systems: fixed MWE identification + parsing

Datasets for experiments I

French Treebank (Abeille et al. 2004)

- dependency version of SPMRL Shared Task 2013 (Seddah et al. 2013)
- MWE annotation modified: regular vs. irregular MWEs (Candito and Constant 2014)
- MWEs limited to compounds (very few verbal expressions)

Streusle Corpus (Schneider et al. 2014)

- Comprehensive annotation of MWEs
- Reviews subpart of the English Web Treebank (Bies et al., 2012)

Datasets for experiments II

Corpus	Streusle		FTB		
	Train	Test	Train	Dev	Test
# sent.	3,312	500	14,759	1,235	2,541
# tokens	48,408	7,171	443,113	38,820	75,216
# MWEs	2,996	401	23,556	2,119	4,043
# fixed	-	-	10,987	925	1,992

Warning: datasets not entirely satisfying

- **FTB**: limited to compounds
- **Streusle**: small datasets

→ Results only provide a partial view

Main experimental findings

Comparison with standard parser with extended labels

- Joint system significantly outperforms it for MWE analysis
- Hard constraints are helpful for syntactic analysis

Comparison with partial systems

- Lexical layer helps syntactic layer prediction
- Syntactic layer does not help lexical layer prediction

Comparison with pipeline system

- Preidentifying fixed MWE is helpful
- Prediction of fixed MWEs seem to confuse non-fixed MWE prediction in joint system

Results on French Treebank

	DEV				TEST			
System	UAS	LAS	MWE	FMWE	UAS	LAS	MWE	FMWE
Extended Labels	86.28	83.67	77.2	83.2	84.85	82.67	75.5	81.9
Ours (explicit)	86.36	83.77	79.7	86.0	84.98	82.79	**79.3**	**84.8**
Ours (implicit)	**86.61**	**84.10**	**80.0**	**86.2**	**85.04**	**82.93**	78.4	84.3
Syntactic only	86.39	83.77	-	85.0	85.02	82.84	-	83.8
Lexical only	-	-	80.0	-	-	-	79.5	-
Fixed only	-	-	-	85.7	-	-	-	85.7
Pipeline	85.49	83.50	**81.8**	85.7	84.84	82.89	**81.1**	**85.7**

Results on Streusle

	TRAIN Cross-validation			TEST		
System	UAS	LAS	MWE	UAS	LAS	MWE
Extended labels	86.16	81.76	49.6	86.31	82.02	46.8
Ours (explicit)	86.25	82.09	52.9	86.05	81.68	**53.4**
Ours (implicit)	**86.81**	**82.68**	**55.0**	**87.05**	**83.14**	51.6
Syntactic only	86.35	82.23	-	86.41	82.20	-
Lexical only	-	-	54.5	-	-	**53.6**

General conclusions

Contributions

- A new representation of lexical and syntactic analysis
- A new transition-based system predicting such representation including special transitions for handling MWEs

Future work

- Implementing more advanced features: beam-search, dynamic oracles, deep learning, distributional semantics
- Evaluating on more relevant datasets (to be built)
- Analysis of produced sequences of transitions (actions)

→ ANR PARSEME-FR Project

Thanks!

Questions/Comments?