Transition-based Parsing of Multiword Expressions

Mathieu Constant
Mathieu.Constant@univ-lorraine.fr

Université de Lorraine
ATILF, CNRS

Joint work with Hazem Al Saied (Univ. Lorraine), Marie Candito (Univ. Paris Diderot), Joakim Nivre (Uppsala University)

Our task

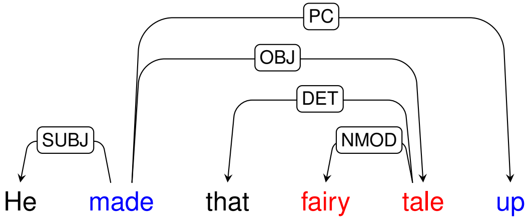
He made that fairy tale up

Our task

He made that fairy tale up

- lexical segmentation (multiword expressions)

Our task



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

This talk

1. Background: multiword expressions (properties, processing)
2. A transition-based system for multiword expression identification
   [Saied et al., 2019]
3. A transition-based system for joint lexical and syntactic analysis
   [Constant and Nivre, 2016]

Multiword Expressions (MWEs) I

Definitional features

- A sequence of multiple lexemes that displays a certain degree of non-compositionality
- i.e. irregularity for one or more linguistic dimensions: morphological, lexical, syntactic, semantic
- Multiword token (*snowman*) vs. multi-token expression (*light house*)

Multiword Expressions (MWEs) II

Examples

- Nominal compounds: *dry run* (rehearsal), *red tape* (excessive bureaucracy)
- Multi-token names: *Los Angeles*
- Adverbial compounds: *above board* (honest), *at all*
- Grammatical complex words: *in spite of*, *as well as*
- Verbal idiomatic expressions: *spill the beans* (reveal), *cut the mustard* (succeed)
- Light verb constructions: *take a shower*
- Verb-particle constructions: *give up*

Multiword expressions

Multidimensional non-compositionality

- **Semantic:** *kick the bucket* (die)
- **(Morpho-)Syntactic:**
  - **irregular pattern:**
    by and large (coordination of a preposition and an adjective)
  - **forbidden expected transformations:**
    *cut the mustard = * the mustard is cut*
    *spill the (beans + *bean)*
- **Lexical:** *cut the (mustard + *mayonnaise)*

MWE challenges for NLP I

Ambiguity

- MWE vs. literal meaning

  *I **looked up** this word* (idiomatic meaning: *search*)
  I looked up to the sky (literal meaning)

- MWE vs. accidental co-occurrence

  ***By and large** we agree* (idiomatic meaning: *overall*)
  He walked by and large tractors passed him (accidental cooccurrence)

Discontiguity

*The dog was **hard on** the young boy**'s heels*** (following very closely)

MWE challenges for NLP II

Non-compositionality

- Various degrees of compositionality

  red tape < green card < traffic light
- Internal compositional modifications

  make** a crucial **decision

Variability

- Morphological variability

  *John **takes** a **nap***
  *John and Mary **take** a **nap***
- Syntactic variability

  *John **made** a **choice***
  *A **choice** was **made** by John*

MWE challenges for NLP III

Nesting

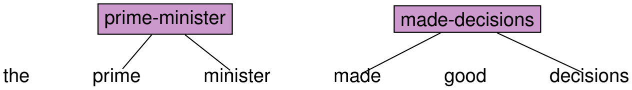
John (took a (rain check))

((Los Angeles) Lakers)

Sharing of components

John $took_{1,2}$ *a* $bath_1$ *then a* $shower_2$

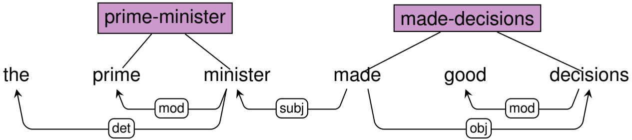
MWE representation



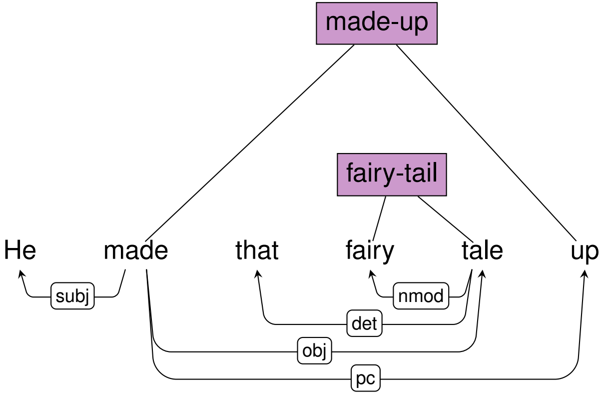
MWE representation

Form: made decisions
Lemma: make decision
POS: V
...

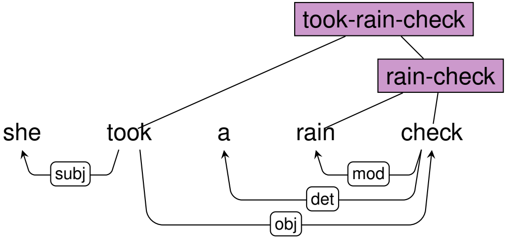
MWE representation



MWE representation (other example)



MWE nesting


took-rain-check
rain-check
she
took
a
rain
check
subj
mod
det
obj

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, binary classification, 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, binary classification, supervised sequential tagging, ...

Supervised sequential tagging I

MWE identification = a standard tagging task

- Each word of a sentence is automatically assigned a tag
- IOB-like tagset
  - B: word is the left-most word of an MWE
  - I: word is inside an MWE, but is not the MWE left-most word.
  - O: word is outside an MWE

[Blunsom and Baldwin, 2006, Constant et al., 2012, Schneider et al., 2014, Maldonado et al., 2017, Taslimipoor and Rohanian, 2018]

Supervised sequential tagging II

Examples

- **Basic tagset**

The	prime	minister	made	good	decisions
O	B	I	B	O	I

- **Extended tagset** [Schneider et al., 2014]

He	made	that	fairy	tail	up
O	B	o	b	i	I

Supervised sequential tagging III

Supervised procedure

- **Training phase**:
  MWE-annotated corpus (with IOB-like scheme) $\rightarrow$ model

- **Identification phase**:
  new tokenized text + model $\rightarrow$ MWE-tagged text

Supervised sequential tagging IV

Types of models used

- **Conditional Random Fields (CRF)** [Blunsom and Baldwin, 2006, Maldonado et al., 2017]
- **Recurrent Neural Networks** [Gharbieh et al., 2017, Klyueva et al., 2017]
- **Convolutional Neural Networks** [Gharbieh et al., 2017]
- **Self-attention** [Rohanian et al., 2019]
- ...

Supervised sequential tagging V

Pros and cons

+ Very easy to implement
+ Efficient

- Limited computational representation (a sequence)
- theoretical coverage limitations: ex. need to use "hacks" to deal with MWEs in gaps

A TRANSITION SYSTEM FOR MULTIWORD EXPRESSION IDENTIFICATION [Saied et al., 2019]

Our approach

- **Use of a parsing mechanism**
  - natural way to deal with gaps (long-distance groupings) and nesting (hierarchical groupings)
  - building of more sophisticated structures (ex. trees): more expressive power
- In particular, **use of transition-based parsing** [Nivre, 2004] ...
- ... **adapted to MWE identification** [Al Saied et al., 2017, Al Saied et al., 2018, Stodden et al., 2018, Saied et al., 2019]

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)

Transition system for MWE identification

- A configuration in our system consists of a triplet $c = (S, B, A)$:
  - S : Stack containing units under processing
  - β : Buffer containing the remaining input tokens
  - A : Set of output VMWEs

SHIFT	$(S, x\|B, A) \Rightarrow (S\|x, B, A)$
REDUCE	$(S\|x, B, A) \Rightarrow (S, B, A)$
MERGE	$(S\|x, y, B, A) \Rightarrow (S\|(x, y), B, A)$
MARK	$(S\|x, B, A) \Rightarrow (S\|x, B, A \cup (x))$

Figure: Set of transitions

Example

He made that fairy tail up

Transition

–

Buffer

[He made that fairy tail up]

Stack

[ ]

MWEs

–

Example

He made that fairy tail up

Transition

Shift

Buffer

[made that fairy tail up]

Stack

[He]

MWEs

–

Example

He made that fairy tail up

Transition

Reduce

Buffer

[made that fairy tail up]

Stack

[]

MWEs

–

Example

He made that fairy tail up

Transition

Shift

Buffer

[that fairy tail up]

Stack

[made]

MWEs

Example

He made that fairy tail up

Transition
Shift

Buffer
[fairy tail up]

Stack
[made that]

MWEs

Example

He made that fairy tail up

Transition
Reduce

Buffer
[fairy tail up]

Stack
[made]

MWEs

Example

He made that fairy tail up

Transition

Shift

Buffer

[tail up]

Stack

[made fairy]

MWEs

Example

He made that fairy tail up

Transition
Shift

Buffer
[up]

Stack
[made fairy tail]

MWEs

Example

He made that fairy tail up

Transition

Merge

Buffer

[up]

Stack

[made fairy-tail]

MWEs

Example

He made that fairy tail up

Transition

Mark

Buffer

[up]

Stack

[made fary-tail]

MWEs

{fary-tail}

Example

He made that fairy tail up

Transition

Reduce

Buffer

[ ]

Stack

[made]

MWEs

{fary-tail}

Example

He made that fairy tail up

Transition

Shift

Buffer

[ ]

Stack

[made up]

MWEs

{fary-tail}

Example

He made that fairy tail up

Transition

Merge

Buffer

[ ]

Stack

[made-up]

MWEs

{fary-tail}

Example

He made that fairy tail up

Transition

Mark

Buffer

[]

Stack

[made-up]

MWEs

{fary-tail, made-up}

Example

He made that fairy tail up

Transition

Reduce

Buffer

[]

Stack

[]

MWEs

{fary-tail, made-up}

Static oracle and greedy parsing algorithm

Oracle used at training time

- Used to construct the training data as a set of (configuration,transition to apply) pairs
- Apply first legal transition compatible with gold data
- **Priority order**: **M**ARK, **M**ERGE, **R**EDUCE, **S**HIFT

At parsing time

- Greedily apply in sequence the best legal local transition
- For each configuration, predict the transition to apply next, **using a classifier**

Task: identification of verbal multi-word expressions

- **Verbal MWEs**
  - Rare
  - Crucial to downstream semantic tasks
  - More difficult to identify than other categories of MWEs
    - More likely to be discontinuous sequences
    - More likely to exhibit morphological and structural variation

- We focus on the task of identifying *verbal* MWEs using PARSEME corpora for open and closed tracks [Ramisch et al., 2018]

- **Objective:** comparing the development and tuning of linear versus neural classifiers

  Support Vector Machine (SVM)
  vs. Multi-Layer Perceptron (MLP)

PARSEME 1.1 Datasets I

Languages

- 20 languages: different families and corpus sizes

Annotations

- several categories: Light verb contructions, idioms, inherently reflexive verbs...
- $\approx 80,000$ annotated verbal MWEs
- size greatly varies across languages

PARSEME 1.1 Datasets II

An MWE =

- A set of tokens
- A single token can be a MWE (e.g. particle+verb in German)
- A few nesting, a few sharing of components (overlapping MWEs)

  make$_{1,2}$ *an adjustment*$_1$ *and an effort*$_2$

This work [Saied et al., 2019]

Constraints

- Use lemmas and POS available in datasets, but not syntactic parses
- Robustness: use the same hyperparameters for all languages

Focus: strategies for hyperparameter tuning

- Resampling techniques
- Vocabulary generalization
- Trend-based hyperparameter tuning

Example

Trans		Configuration $= (S, B, A)$
$F_i(s)$	$\Rightarrow$	$[\,], [\text{Take}, .., \text{account}], [\,]$
SHIFT	$\Rightarrow$	$[\text{Take}], [\text{the}, .., \text{account}], [\,]$
SHIFT	$\Rightarrow$	$[\text{Take}, \text{the}], [\text{fact}, .., \text{account}], [\,]$
REDUCE	$\Rightarrow$	$[\text{Take}], [\text{fact}, .., \text{account}], [\,]$
SHIFT	$\Rightarrow$	$[\text{Take}, \text{fact}], [\text{that}, .., \text{account}], [\,]$
...		
SHIFT	$\Rightarrow$	$[\text{Take}, \text{give}], [\text{up}, \text{into}, \text{account}], [\,]$
SHIFT	$\Rightarrow$	$[\text{Take}, \text{give}, \text{up}], [\text{into}, \text{account}], [\,]$
MERGE	$\Rightarrow$	$[\text{Take}, (\text{give}, \text{up})], [\text{into}, \text{account}], [\,]$
MARK	$\Rightarrow$	$[\text{Take}, (\text{give}, \text{up})], [\text{into}, \text{account}], [(\text{give}, \text{up})]$
REDUCE	$\Rightarrow$	$[\text{Take}], [\text{into}, \text{account}], [(\text{give}, \text{up})]$
SHIFT	$\Rightarrow$	$[\text{Take}, \text{into}], [\text{account}], [(\text{give}, \text{up})]$
MERGE	$\Rightarrow$	$[(\text{Take}, \text{into})], [\text{account}], [(\text{give}, \text{up})]$
SHIFT	$\Rightarrow$	$[(\text{Take}, \text{into}), \text{account}], [\,], [(\text{give}, \text{up})]$
MERGE	$\Rightarrow$	$[((\text{Take}, \text{into}), \text{account})], [\,], [(\text{give}, \text{up})]$
MARK	$\Rightarrow$	$[((\text{Take}, \text{into}), \text{account})], [\,], (\text{give}, \text{up}), ((\text{Take}, \text{into}), \text{account})]$
REDUCE	$\Rightarrow$	$[\,], [\,], [(\text{give}, \text{up}), ((\text{Take}, \text{into}), \text{account})]$

Figure: Transition sequence for the sentence *Take*$_1$ *the fact that I didn't give*$_2$ *up*$_2$ *into*$_1$ *account*$_1$.

Tuning methodology I

Preliminaries

- Three pilot languages (BG, PT, TR)
- Limiting training sets to the average size across languages (270*k* tokens)

Random search [Bergstra and Bengio, 2012]

- proved to be more efficient than grid search with the same computational budget
- 1000 trials for each model, on each track (closed / open)
- MLP: Result of each trial is the avg of 2 runs (with seed 1 and 0)

Tuning methodology II

Selecting hyperparameter sets

1. BoRdm: the best performing hyperparameter set in random search
2. Trend-based strategy

   Hyperparameter value = the observed trend among the top k best configurations

Linear model (SVM) - Feature templates

Tuning	BoRdm	TrendB	Feature template
Prelim	+	+	Unigrams S_0, S_1, B_0
Prelim	+	+	Bigrams $S_0S_1, S_0B_0, S_0B_1, S_1B_0$
Prelim	+	+	S_0 in MWT dictionary
Prelim	-	-	Resampling
Rdm Search	-	-	word forms ngrams
Rdm Search	+	-	Unigram B_1
Rdm Search	+	-	Bigram S_0B_2
Rdm Search	+	-	Trigram $S_1S_0B_0$
Rdm Search	+	+	Distance between S_0 and S_1, S_0 and B_0
Rdm Search	+	-	MWE component dictionary
Rdm Search	-	-	Stack length
Rdm Search	+	+	Transition history (length 1)
Rdm Search	-	+	Transition history (length 2)
Rdm Search	+	-	Transition history (length 3)

Table: Feature templates and their values in the the best performing hyperparameter set **BoRdm** and in Trend-based hyperparameter set **TrendB**.

MLP model

Plain feed-forward network

- Embedding layer concatenating the embeddings for the POS of S_0, S_1, B_0 and B_1 and for either their word form or lemma
- Dense layer with ReLU activation, connected to a softmax layer

Vocabulary

- **Exhaustive vocabulary**: hapaxes are replaced at training time by a UNK symbol, with probability 0.5;
- **Compact vocabulary**: any token whose lemma is never a component of a MWE in the training set is replaced by UNK

Resampling I

Preliminary remarks

- Preliminary experiments without resampling showed unstable loss and rather low performance
- Very skewed class distribution for transitions

Resampling techniques (performed systematically)

- **Under sampling:** removes training sentences not containing any MWE
- **Random oversampling**: forces a uniform distribution of the classes by randomly duplicating minority instances

Resampling II

Additional techniques (but proved ineffective by tuning)

- **Focused oversampling:** aims at mimicking a minimum number of occurrences for all MWEs
- **Over loss:** penalizes the model when it fails to predict MERGE and MARK, by multiplying the loss by a coefficient

MLP model - tuning

Type	Hyperparameter	Range or set	BoRdm$_c$	BoRdm$_o$	TB
Embedding and initialisation	Use B_2	{True, False}	True	True	True
	Use B_{-1}	{True, False}	True	False	True
	Lemmatization	{True, False}	True	True	True
	Word emb size	[100, 600]	157	300	300
	POS emb size	[15, 150]	147	132	35
	Pre-trained	{True, False}	False	True	True
	Trainable	{True, False}	True	True	True
	Averaging	{True, False}	False	True	True
	Compact Vocab	{True, False}	True	False	True
Dense	Unit number	[25, 600]	85	56	75
	Dropout	{.1, .2,.. .6}	0.3	0.1	0.4
Sampling	Focused	{True, False}	False	False	False
	Over loss	{True, False}	False	False	False
Train	Learning rate	[.01, .2]	0.017	0.095	0.03
	Batch size	{16, 32, 48, 64, 128}	128	16	48

Table: MLP hyperparameters their possible values in tuning **(Range or set)** , and their optimal values in Best-of-random closed (**BoRdm$_c$**), Best-of-random open (**BoRdm$_o$**) and Trend-based values (**TB**).

Results on test sets

Best ST1.1 system in closed track: TRAVERSAL [Waszczuk, 2018]
Best ST1.1 system in open track: SHOMA
[Taslimipoor and Rohanian, 2018]

Language	Closed track			Open track	
	SVM	MLP_c	ST1.1	MLP_o	ST1.1
DE	49.5	**51.5**	45.3	**49.9**	45.5
F_G	60.8	**62.6**	57.8	**62.3**	58.7
F_G best sys			54.0		58.1

Table: MWE-based F-scores for ST1.1 languages on test sets using our tuned **SVM** and **MLP** models. **ST1.1** = best scores in shared task (artificial score, with best scores for each language in closed and open tracks / best system)

Findings I

Global scores

- Linear versus neural classifiers used in a transition system for MWE identification
  - We long thought SVM would win
  - but the MLP won in the (much more difficult to tune)
- New **state-of-the art scores** on the PARSEME 1.1 shared task data sets;

  but [Rohanian et al., 2019] obtain very high results on 4 languages, using ELMO embeddings and advanced neural architecture

Findings II

Specific findings

- Our **Trend-based tuning methodology** proved beneficial;
- **Class balancing** proved crucial for MLP models;
- **Generalization power** quite low: F-score=12 for SVM and 0 for MLP on unseen MWEs;
- **Pre-trained word embeddings** did not bring any improvement.
- **Future work**: focus on discovery of unseen MWEs, contextualized embeddings

A TRANSITION SYSTEM FOR JOINT LEXICAL AND SYNTACTIC ANALYSIS [Constant and Nivre, 2016]

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)

A few words on data-driven dependency parsing

- Mainly no underlying grammatical formalism
- Parsing algorithms vary from local search (Nivre 2003) to global search (McDonald 2005)
- Use of machine learning techniques: the deep learning revolution (Chen and Manning 2014, Dyer 2015, Weiss et al. 2015, Kiperwasser and Goldberg 2016, Dozat and Manning 2017, ...)

A few words on data-driven dependency parsing

- Mainly no underlying grammatical formalism
- Parsing algorithms vary from local search (Nivre 2003) to global search (McDonald 2005)
- Use of machine learning techniques: the deep learning revolution (Chen and Manning 2014, Dyer 2015, Weiss et al. 2015, Kiperwasser and Goldberg 2016, Dozat and Manning 2017, ...)

Data-driven MWE-aware parsing Orchestration

Three positions for MWE identification

- **Before parsing**: retokenization (*carte verte* → *carte_verte*)
  - gold (Nivre 2004, Arun 2005)
  - predicted (Cafferkey 2007, Constant et al. 2012)
- **During parsing**: joint approach
  - Standard parsers (Nivre et Nilsson 2004, Arun and Keller 2005, ...)
  - Multilayer parsers (Constant et al. 2016, Constant and Nivre 2016)
  - Multitasking (Taslimipoor and Rohanian 2019)
- **After parsing**: Performing MWE identification on parsed text

  (Fazly et al. 2009, Maldonado et al. 2017, Al Saied et al. 2017, Waszczuk 2018, Wasczuk et al 2019)

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

Joint approach using standard dependency parsers

Principle

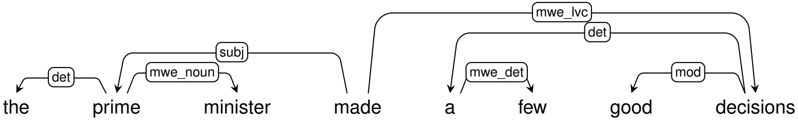
- **Each MWE is annotated as a subtree** of the syntactic tree in the reference treebank
- Use of **off-the-shelf parsers** that are learned from the reference treebank

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 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, 2014)

Irregular MWEs

- They display irregular syntactic structure (e.g., *by and large* = Prep Conj Adj)
- Use of flat MWE representation

→ fixed MWEs in Constant and Nivre (2016)

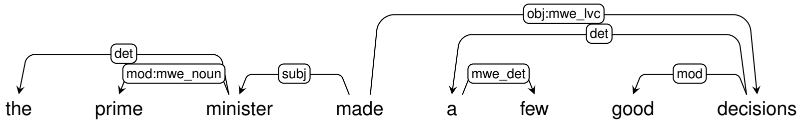
Regular MWEs

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

→ non-fixed MWEs in Constant and Nivre (2016)

A dual MWE representation II

(Candito and Constant, 2014)



Multilayer lexical and syntactic parsing

Drawback of standard parsers

- Hard to represent MWE nesting
- |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

Our contributions
(Constant and Nivre 2016)

A new factorized representation of lexical and syntactic analysis

- Dependency analysis
- Inclusion of Multiword Expression analysis

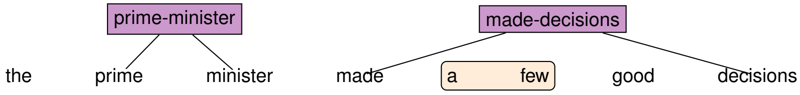
A new transition-based system

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

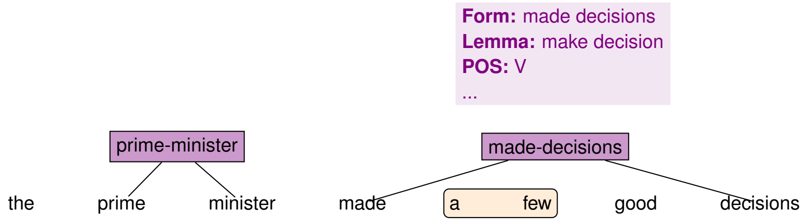
Lexical and Syntactic representation

the prime minister made a few good decisions

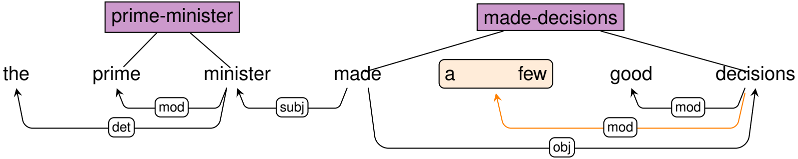
Lexical and Syntactic representation



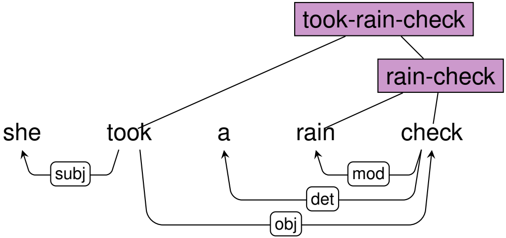
Lexical and Syntactic representation



Lexical and Syntactic representation



MWE embedding



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

$([\,], [x]$, SynArcs, $[\,]$, 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

Merge_F(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

Merge_N(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

$\rightarrow$ 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

General conclusions

Transition-based systems

- Permit tree-structured representation of multiword expressions
- Possibility to perform jointly lexical and syntactic analylis
- Efficient in practice
- Implementation is less straightforward than sequence tagging, but still feasible

Future work

- better handling of unseen multiword expressions (semi-supervised approaches, distributional semantics)
- Implementing more advanced features: beam-search, dynamic oracles, contectualized embeddings, self-attention

Thanks!

Questions/Comments?

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**

References I

Al Saied, H., Candito, M., and Constant, M. (2017).
TheATILF-LLF system for parseme shared task: a transition-based verbal multiword expression tagger.
In *Proceedings of the 13th Workshop on Multiword Expressions (MWE)- 2017*, pages 127–132, Valencia, Spain. Association for Computational Linguistics.

Al Saied, H., Candito, M., and Constant, M. (2018).
A transition-based verbal multiword expression analyzer.
In *Multiword expressions at length and in depth: Extended papers from the MWE 2017 workshop*, volume 2, page 209. Language Science Press.

Bergstra, J. and Bengio, Y. (2012).
Random search for hyper-parameter optimization.
Journal of Machine Learning Research, 13(Feb):281–305.

References II

Blunsom, P. and Baldwin, T. (2006).
Multilingual deep lexical acquisition for hpsgs via supertagging.
In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 164–171. Association for Computational Linguistics.

Constant, M. and Nivre, J. (2016).
A transition-based system for joint lexical and syntactic analysis.
In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 161–171.

References III

Constant, M., Sigogne, A., and Watrin, P. (2012).
Discriminative strategies to integrate multiword expression recognition and parsing.
In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 204–212. Association for Computational Linguistics.

Gharbieh, W., Bhavsar, V., and Cook, P. (2017).
Deep learning models for multiword expression identification.
In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 54–64.

Klyueva, N., Doucet, A., and Straka, M. (2017).
Neural networks for multi-word expression detection.
In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, pages 60–65, Valencia, Spain. Association for Computational Linguistics.

References IV

Maldonado, A., Han, L., Moreau, E., Alsulaimani, A., Chowdhury, K. D., Vogel, C., and Liu, Q. (2017).
Detection of verbal multi-word expressions via conditional random fields with syntactic dependency features and semantic re-ranking.
In *Proceedings of the 13th Workshop on Multiword Expressions(MWE 2017)*, pages 114–120, Valencia, Spain. Association for Computational Linguistics.

Nivre, J. (2004).
Incrementality in deterministic dependency parsing.
In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57. Association for Computational Linguistics.

References V

Ramisch, C., Cordeiro, S., Savary, A., Vincze, V., Mititelu, V., Bhatia, A., Buljan, M., Candito, M., Gantar, P., Giouli, V., et al. (2018).
Edition 1.1 of the parseme shared task on automatic identification of verbal multiword expressions.
In *the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 222–240.

Rohanian, O., Taslimipoor, S., Kouchaki, S., Ha, L. A., and Mitkov, R. (2019).
Bridging the gap: Attending to discontinuity in identification of multiword expressions.
arXiv preprint arXiv:1902.10667.

References VI

Saied, H. A., Candito, M., and Constant, M. (2019).
Comparing linear and neural models for competitive MWE identification.
In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 86–96, Turku, Finland. Linköping University Electronic Press.

Schneider, N., Danchik, E., Dyer, C., and Smith, N. A. (2014).
Discriminative lexical semantic segmentation with gaps: running the mwe gamut.
Transactions of the Association for Computational Linguistics, 2:193–206.

References VII

Stodden, R., QasemiZadeh, B., and Kallmeyer, L. (2018).
Trapacc and trapaccs at parseme shared task 2018: Neural transition tagging of verbal multiword expressions.
In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 268–274.

Taslimipoor, S. and Rohanian, O. (2018).
Shoma at parseme shared task on automatic identification of vmwes: Neural multiword expression tagging with high generalisation.
arXiv preprint arXiv:1809.03056.

References VIII

Waszczuk, J. (2018).
Traversal at parseme shared task 2018: Identification of verbal multiword expressions using a discriminative tree-structured model.
In *Proceedings of the Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions (LAW-MWE-CxG-2018)*, pages 275–282.