Experiments in Learning to Solve Formal Analogical Equations

Rafik Rhouma and Philippe Langlais

Dept. I.R.O.
Université de Montréal, Québec, Canada
felipe@iro.umontreal.ca

ICCBR 2018

Analogy

$[x : y :: z : t]$ stands for a proportional analogy

- x is to y as z is to t


:

::

:


Analogy

$[x : y :: z : t]$ stands for a proportional analogy

- x is to y as z is to t


:

::

:


Formal Analogies (or Analogies on Forms)

- $[x \setminus y \bigcirc z \supset t]$ denotes a formal analogy
  1. $[funny \setminus funniest \bigcirc lucky \supset luckiest]$
  2. $[suggère \setminus suggérer \bigcirc ingère \supset ingérer]$
  3. $[This\ guy\ drinks\ too\ much \setminus This\ boat\ sinks \bigcirc These\ guys\ drank\ too\ much \supset These\ boats\ sank]$
- Several operational definitions of formal analogy ***[?, ?]***

Definition of *[?]*

$[x \setminus y \bigcirc z \supset t]$ **iff** we can find d**-factorizations** f_x, f_y, f_z and f_t such that, $\forall i \in [1, d]$:

$$(f_y^{(i)}, f_z^{(i)}) \in \left\{(f_x^{(i)}, f_t^{(i)}), (f_t^{(i)}, f_x^{(i)})\right\}$$

- $[$*this guy drinks too much* $\setminus$ *this boat sinks* $\bigcirc$ *these guys drank too much* $\supset$ *these boats sank*$]$ because:

x	$\equiv$	this	guy	ϵ	dr	inks	too much
y	$\equiv$	this	boat	ϵ	s	inks	ϵ
z	$\equiv$	these	guy	s	dr	ank	too much
t	$\equiv$	these	boat	s	s	ank	ϵ

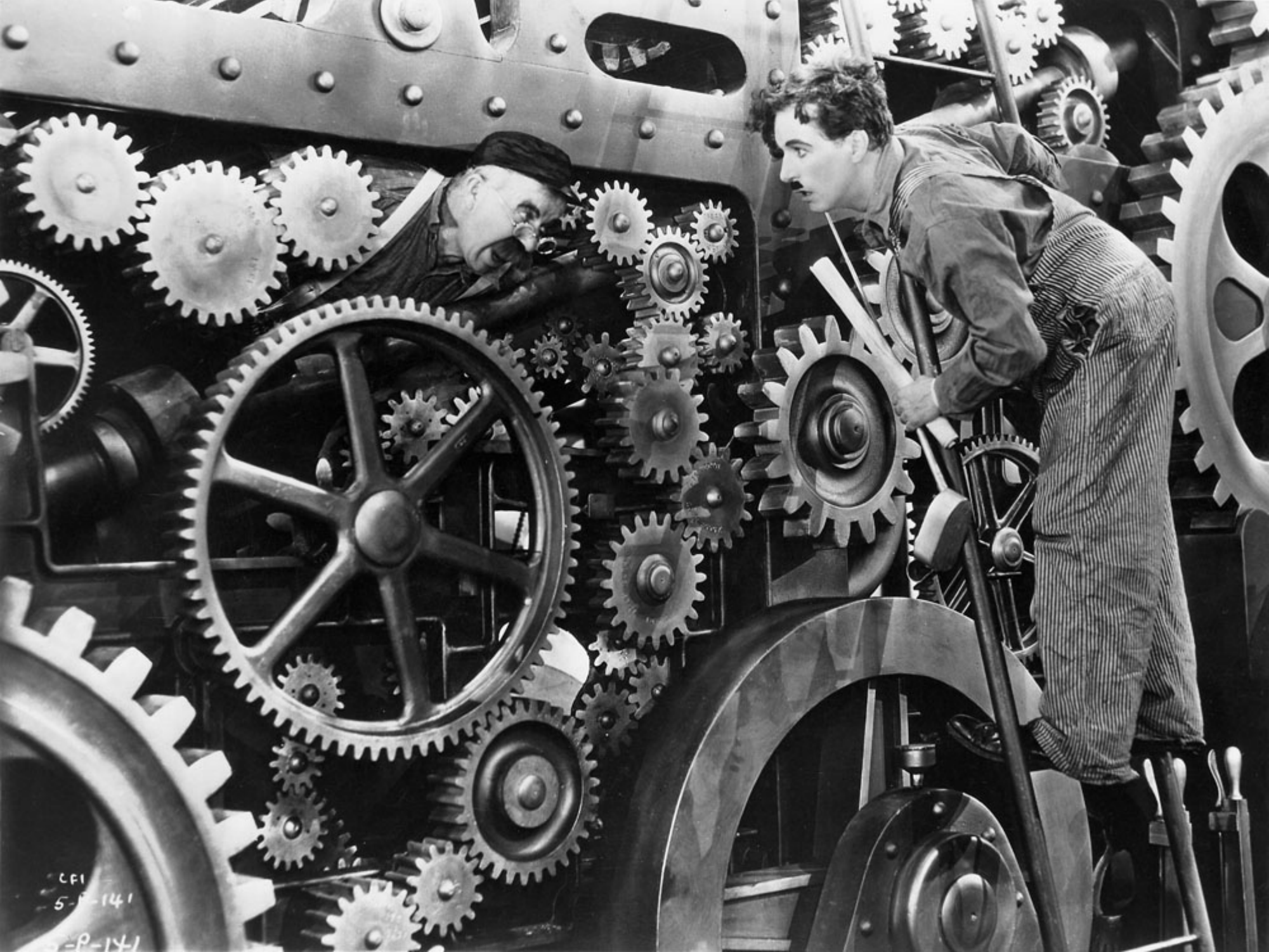
Definition of *[?]*

$[x \setminus y \bigcirc z \supset t]$ **iff** we can find d**-factorizations** f_x, f_y, f_z and f_t such that, $\forall i \in [1, d]$:

$$(f_y^{(i)}, f_z^{(i)}) \in \left\{(f_x^{(i)}, f_t^{(i)}), (f_t^{(i)}, f_x^{(i)})\right\}$$

- $[$*this guy drinks too much* $\setminus$ *this boat sinks* $\bigcirc$ *these guys drank too much* $\supset$ *these boats sank*$]$ because:

x	$\equiv$	this	guy	ϵ	dr	inks	too much
y	$\equiv$	this	boat	ϵ	s	inks	ϵ
z	$\equiv$	these	guy	s	dr	ank	too much
t	$\equiv$	these	boat	s	s	ank	ϵ


CFI
5-P-141

(Yvon & al., 2004)

$$[x \setminus y \bigcirc z \supset t] \iff t \in (y \bullet z) \setminus x$$

- **shuffle** $a \bullet b$ read sequences in a and b from left to right, allowing to switch from one string to another

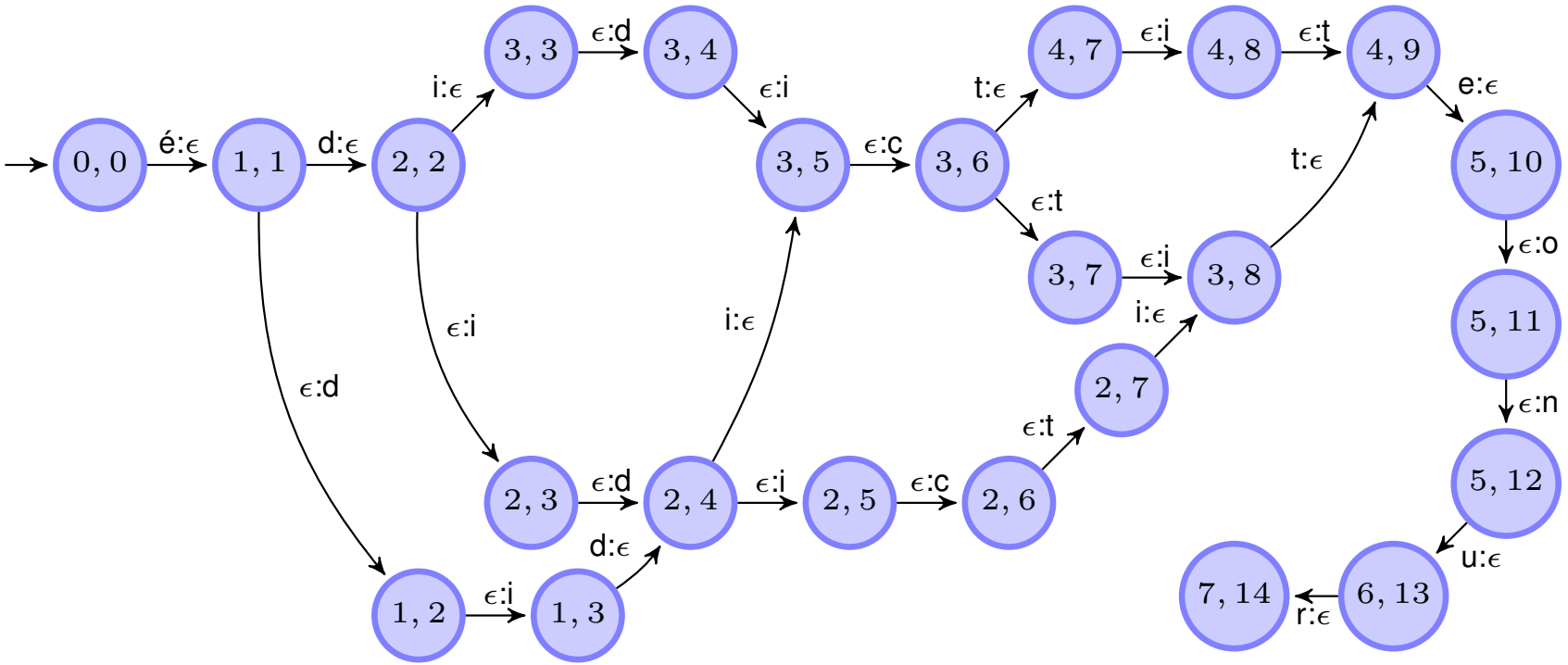
  $spondyondontilalgiatis$ et $ondspondonylaltitisgia$
  $\in spondylalgia \circ ondontitis$

- **complement** $a \setminus b$ strings obtained by removing substring b in a

  $spondylitis \in spondyondontilalgiatis \setminus ondontalgia$
  $spydoniltis \in spondyondontilalgiatis \setminus ondontalgia$

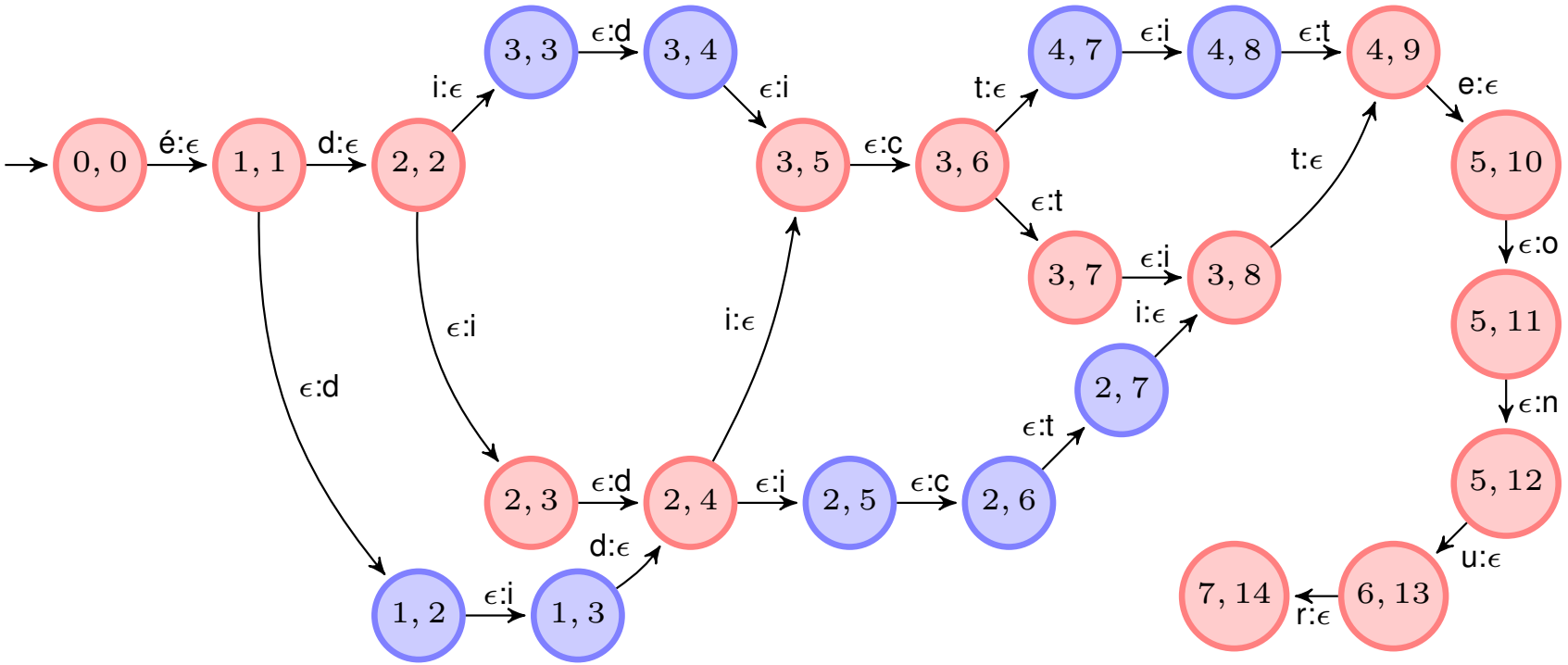
$\{(y \bullet z) \setminus x\}$ is a rational language

édition • dicteur \ éditeur



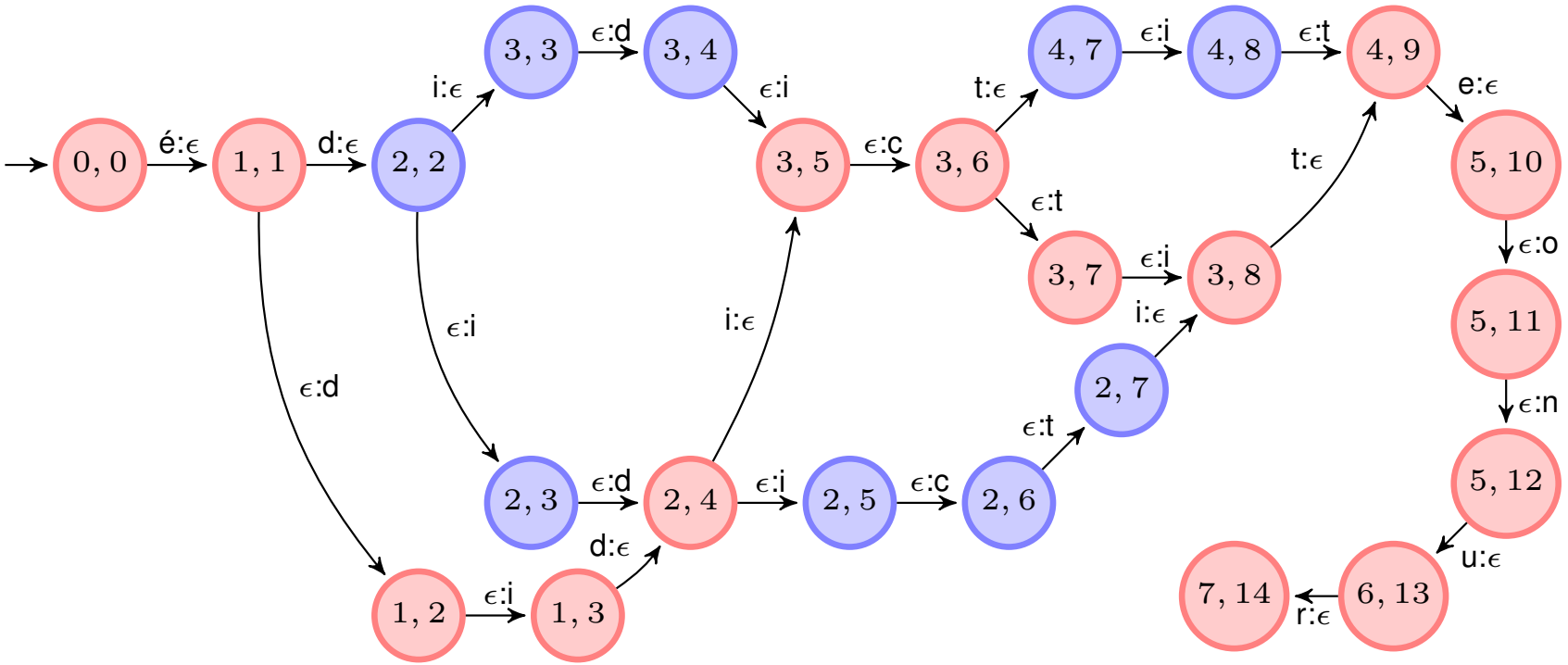
idction , diction , idciton, diicton, diciton

édition • dicteur \ éditeur



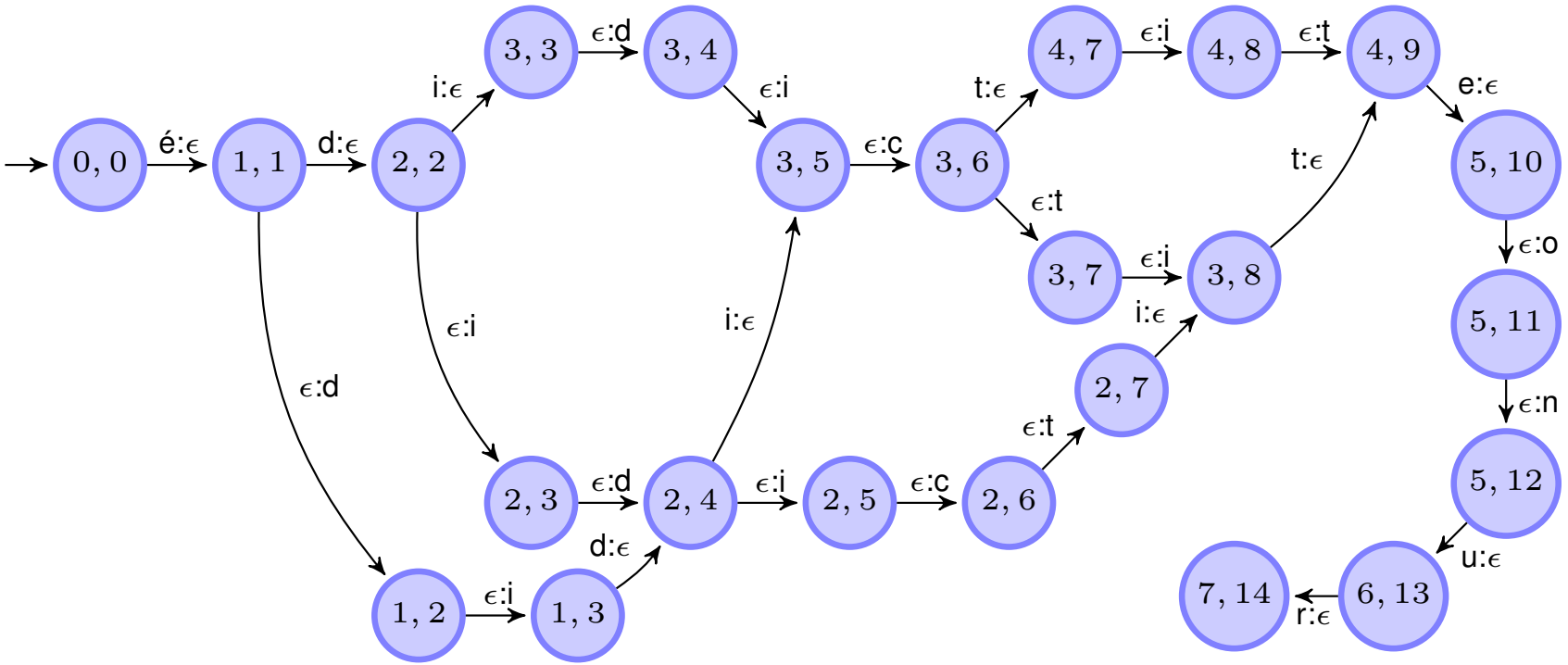
idction , diction , idciton, diicton, diciton

édition • dicteur \ éditeur



idction , diction , idciton, diicton, diciton

édition • dicteur \ éditeur



idction , diction , idciton, diicton, diciton

Solver alea *[?]*

- building the automaton typically is combinatorial
- alea randomly samples ρ shuffles of y and z and complements them (with x)

- **notes:**
  - for too low values of ρ, alea may fail to identify a solution
  - different shuffles may lead (after complementation) to the same solution
  - solutions are ranked in decreasing order of frequency

[*this guy drinks too much* $\setminus$ *this boat sinks* $\bigcirc$ *those guys drink too much* $\supset$ *?*]

$\rho = 20$	$n = 8$
$t = 0.0003$	$rank = \phi$
thos_boate_sinks (2)	
tho_boatse_sinks (2)	
thoboatse__sinks (2)	

$\rho = 100$	$n = 28$
$t = 0.001$	$rank = 13$
thoboatse__sinks (2)	
tho_boatse_sinks (2)	
those_sboat_sink (2)	

$\rho = 1000$	$n = 28$
$t = 0.009$	$rank = 2$
those_boat_ssink (5)	
those_boats_sink (5)	
thoes_tboa_sinks (5)	

$\rho = 10^6$	$n = 19\,796$
$t = 3.82$	$rank = 10$
thoes_boat_sinks (2550)	
thoses_boat_sink (1037)	
those_boat_ssink (999)	

- ρ: nb of shuffles sampled
- n: # of solutions proposed
- $rank$: rank of the reference solution ($\phi \equiv$ not found)
- t: time (in sec.)

[*this guy drinks too much* $\setminus$ *this boat sinks* $\bigcirc$ *those guys drink too much* $\supset$ *?*]

$\rho = 20$	$n = 8$
$t = 0.0003$	$rank = \phi$
thos_boate_sinks (2)	
tho_boatse_sinks (2)	
thoboatse__sinks (2)	

$\rho = 100$	$n = 28$
$t = 0.001$	$rank = 13$
thoboatse__sinks (2)	
tho_boatse_sinks (2)	
those_sboat_sink (2)	

$\rho = 1000$	$n = 28$
$t = 0.009$	$rank = 2$
those_boat_ssink (5)	
those_boats_sink (5)	
thoes_tboa_sinks (5)	

$\rho = 10^6$	$n = 19\,796$
$t = 3.82$	$rank = 10$
thoes_boat_sinks (2550)	
thoses_boat_sink (1037)	
those_boat_ssink (999)	

- ρ: nb of shuffles sampled
- n: # of solutions proposed
- $rank$: rank of the reference solution ($\phi \equiv$ not found)
- t: time (in sec.)

[*this guy drinks too much* $\setminus$ *this boat sinks* $\bigcirc$ *those guys drink too much* $\supset$ *?*]

$\rho = 20$	$n = 8$
$t = 0.0003$	$rank = \phi$
thos_boate_sinks (2)	
tho_boatse_sinks (2)	
thoboatse__sinks (2)	

$\rho = 100$	$n = 28$
$t = 0.001$	$rank = 13$
thoboatse__sinks (2)	
tho_boatse_sinks (2)	
those_sboat_sink (2)	

$\rho = 1000$	$n = 28$
$t = 0.009$	$rank = 2$
those_boat_ssink (5)	
those_boats_sink (5)	
thoes_tboa_sinks (5)	

$\rho = 10^6$	$n = 19\,796$
$t = 3.82$	$rank = 10$
thoes_boat_sinks (2550)	
thoses_boat_sink (1037)	
those_boat_ssink (999)	

- ρ: nb of shuffles sampled
- n: # of solutions proposed
- $rank$: rank of the reference solution ($\phi \equiv$ not found)
- t: time (in sec.)

[*this guy drinks too much* $\setminus$ *this boat sinks* $\bigcirc$ *those guys drink too much* $\supset$ *?*]

$\rho = 20$	$n = 8$
$t = 0.0003$	$rank = \phi$
thos_boate_sinks (2)	
tho_boatse_sinks (2)	
thoboatse__sinks (2)	

$\rho = 100$	$n = 28$
$t = 0.001$	$rank = 13$
thoboatse__sinks (2)	
tho_boatse_sinks (2)	
those_sboat_sink (2)	

$\rho = 1000$	$n = 28$
$t = 0.009$	$rank = 2$
those_boat_ssink (5)	
those_boats_sink (5)	
thoes_tboa_sinks (5)	

$\rho = 10^6$	$n = 19\,796$
$t = 3.82$	$rank = 10$
thoes_boat_sinks (2550)	
thoses_boat_sink (1037)	
those_boat_ssink (999)	

- ρ: nb of shuffles sampled
- n: # of solutions proposed
- $rank$: rank of the reference solution ($\phi \equiv$ not found)
- t: time (in sec.)

Solver word2vec

- $queen - king \sim woman - man$ **[?]**
- Solve $[x : y :: z : ?]$ by:

$$\hat{t} = \underset{t \in V}{\operatorname{argmax}} \cos(t, z - x + y)$$

- Can also solve syntactic equations:
  $[work \setminus worked \bigcirc accept \supset ?]$ ▷ accepted
- **Note:** rank words in V but does not generate new ones

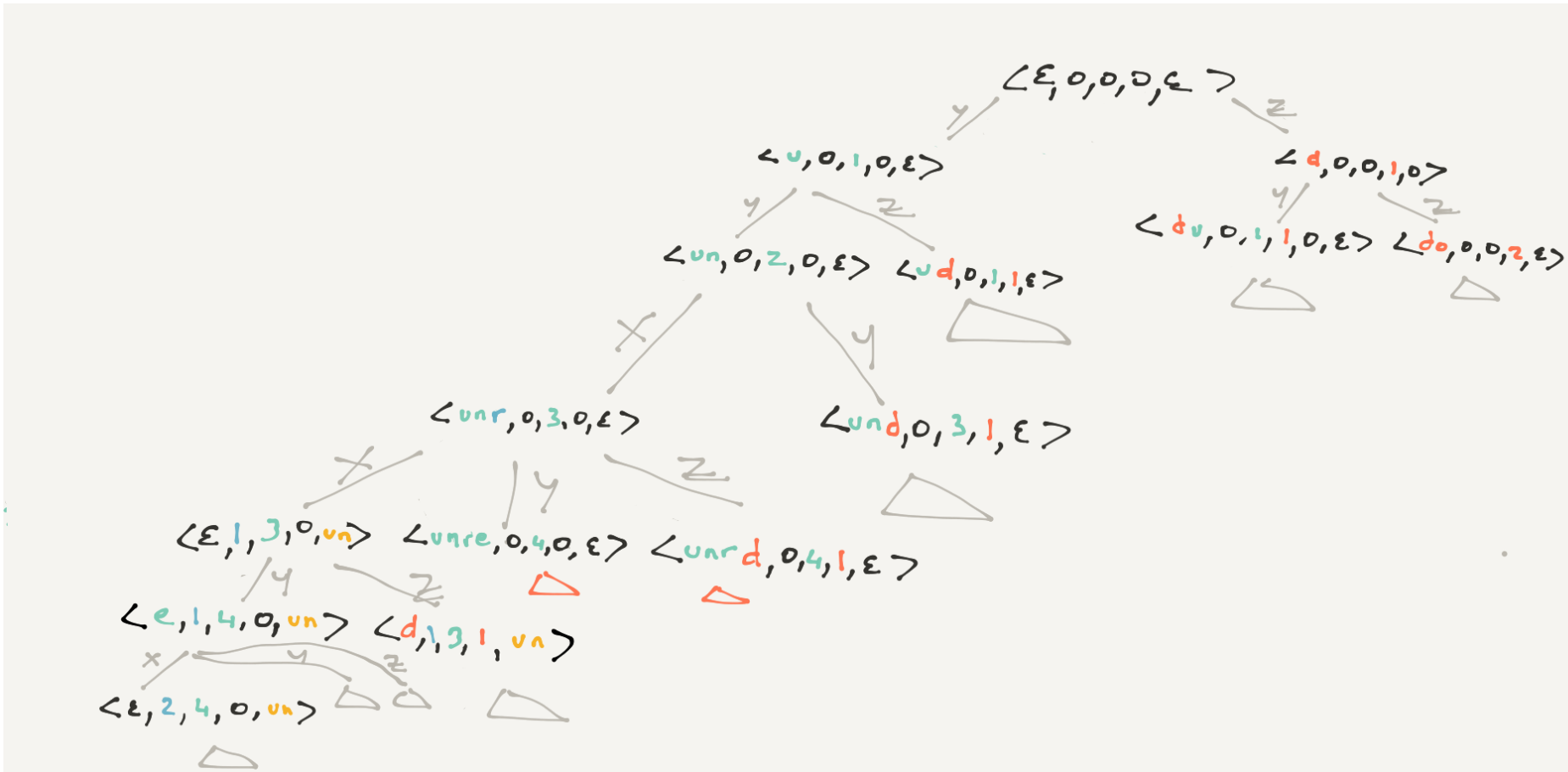
Learning to Solve Equations

- making use of a training set of equations and their solutions $\mathcal{T} = \{((x, y, z), t)\}$ where $[x \setminus y \bigcirc z \supset t]$
- using structured learning
  - given $g : \mathcal{I}^3 \times \mathcal{I} \to \mathbb{R}$ which evaluates a fit between a triple of strings in $\mathcal{I}$, $i \equiv (x, y, z)$, and any string t, we seek to find (search):

$$\hat{t} = \underset{t \in \mathcal{I}}{\operatorname{argmax}}\, g(i, t)$$

  - we assume a linear model for $g = \langle \mathbf{w}, \Phi(i, t) \rangle$ parametrized by a feature vector $\mathbf{w}$ in $\mathbb{R}^K$ and a feature map $\Phi(i, t)$ decomposed into K binary feature fonctions $\phi_k : (i, t) \to \{0, 1\}$.
  - we wish to adjust $\mathbf{w}$ so as to minimize over $\mathcal{T}$ the number of search errors, thanks to the voted perceptron algorithm ***[?]***

Search


⟨ε,0,0,0,ε⟩
⟨u,0,1,0,ε⟩
⟨d,0,0,1,0⟩
⟨un,0,2,0,ε⟩
⟨ud,0,1,1,ε⟩
⟨dv,0,1,1,0,ε⟩
⟨do,0,0,2,ε⟩
⟨unr,0,3,0,ε⟩
⟨und,0,3,1,ε⟩
⟨ε,1,3,0,un⟩
⟨unre,0,4,0,ε⟩
⟨unrd,0,4,1,ε⟩
⟨e,1,4,0,un⟩
⟨d,1,3,1,un⟩
⟨ε,2,4,0,un⟩

Peculiarities of the Search

- Tree of depth $|x| + |y| + |z|$ with a branching factor close to 2 on average (too many nodes)
- Only few actions increase the prefix of a solution (in particular, many states with an empty prefix)

In practice, we:

- search space organized as a graph
- check that complementation is still possible
- control the maximum number of X or Y actions that can take place without complementing with x

Feature Map

3 families of binary features for characterizing $< s, i, j, k, p >$:

language model (14 features) evaluating the likelihood of the prefix p (or shuffle s) so far, according to a n-gram LM trained on an out-domain monolingual corpus.

- $\exists ?i : p_{LM}(p_i|p_{i-2}p_{i-1}) < \delta$

edit-distance (20 features) to enforce that the solution shares with y and z subsequences

- [*reader* : *unreadable* :: *doer* : *undoable*]

search-based (20k features)

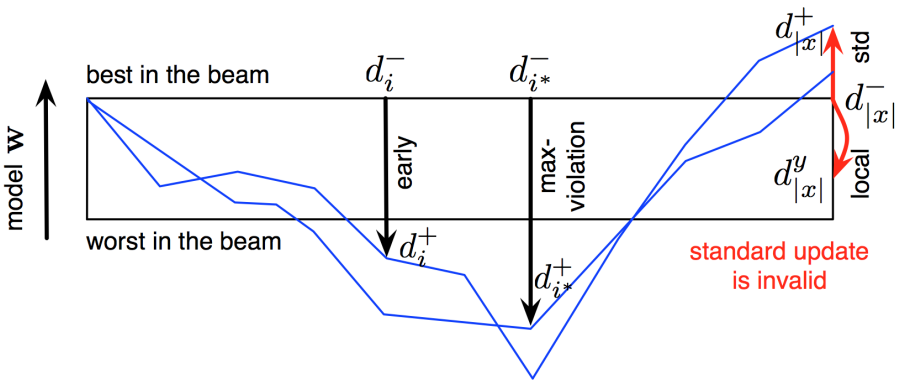
- global: e.g. number of Y or Z actions in a row
- local: e.g. $(x_i, y_j, z_k) \equiv (r, a, d)$

Averaged Voted Perceptron *[?]*

$\mathbf{w}, \mathbf{w}_a \leftarrow \mathbf{0}$
$e \leftarrow 0$
repeat
  $e \leftarrow e + 1$
  for all example $(i, t) \in D$ **do**
    $\hat{t} = \text{argmax}_t \, \mathbf{w}^T \Phi(i, t)$
    if $\hat{t} \neq t$ **then**
      $\mathbf{w} \leftarrow \mathbf{w} + \Phi(i, t) - \Phi(i, \hat{t})$
      $\mathbf{w}_a \leftarrow \mathbf{w}_a + \mathbf{w}$
until converged
return $\mathbf{w}_a / e.|\mathcal{D}|$

Averaged Voted Perceptron Variants

- Not suited for inexact search (our case) ***[?]***
  - update **only if** $\Phi(i,t) - \Phi(i,\hat{t}) < 0$
- We implemented 4 different variants (using forced-decoding)



standard do not care

safe remove the example if condition is not satisfied

early update on a prefix whenever there is no prefix of a reference solution in the beam

late update on the largest prefix that satisfies the condition

Metrics

accuracy	percentage of test equations for which the solver output the correct solution at rank 1
silence	percentage of test equations without any solution proposed

Dataset `google`

- Formal analogies of the dataset released by [**?**]

# of analogies: 4977		word's avr. length: 7
adjective-adverbe	ADJ-ADV	[*amazing* \ *amazingly* ◯ *serious* ⊃ *seriously*]
opposite	OPP	[*certain* \ *uncertain* ◯ *competitive* ⊃ *uncompetitive*]
comparative	COMP	[*fast* \ *faster* ◯ *bright* ⊃ *brighter*]
superlative	SUP	[*warm* \ *warmest* ◯ *strange* ⊃ *strangest*]
present-participle	PP	[*code* \ *coding* ◯ *dance* ⊃ *dancing*]
nationality-adverb	NAT	[*Australia* \ *Australian* ◯ *Croatia* ⊃ *Croatian*]
past-tense	PAST	[*decreasing* \ *decreased* ◯ *listening* ⊃ *listened*]
plural	PLUR	[*eye* \ *eyes* ◯ *donkey* ⊃ *donkeys*]
plural-verbs	PL-VB	[*listen* \ *listens* ◯ *eat* ⊃ *eats*]

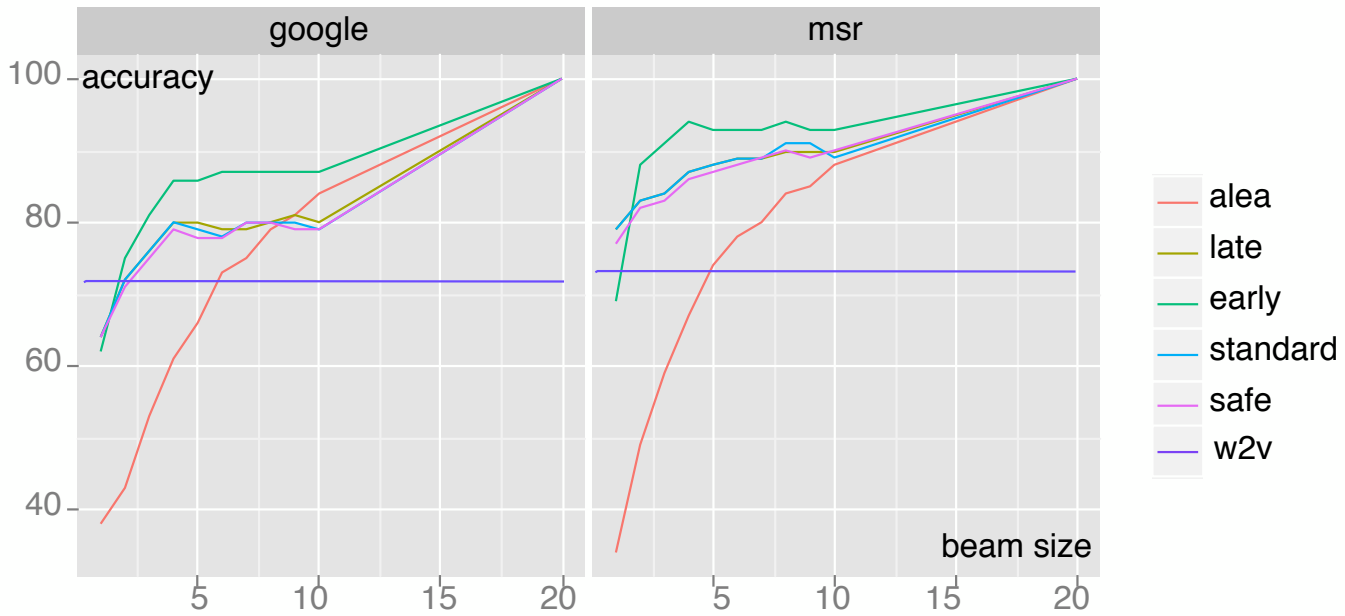
Dataset: `msr`

- Formal analogies of the dataset released by [**?**]

# of analogies: 3664	word's avr. length: 6
JJ-JJR	$[high \setminus higher \bigcirc wild \supset wilder]$
JJR-JJ	$[greater \setminus greatest \bigcirc earlier \supset earliest]$
JJS-JJ	$[low \setminus lowest \bigcirc short \supset shortest]$
NN-NNPOS	$[problem \setminus problems \bigcirc program \supset programs]$
VB-VBP	$[take \setminus takes \bigcirc run \supset runs]$
VB-VBD	$[prevent \setminus prevented \bigcirc consider \supset considered]$
NNPOS-NN	$[days \setminus day \bigcirc citizens \supset citizen]$
VBZ-VBD	$[believes \setminus believed \bigcirc likes \supset liked]$

Accuracy as a function of the beam size (or ρ)

- training on `msr` and testing on `google` (left), or the other way round (right)

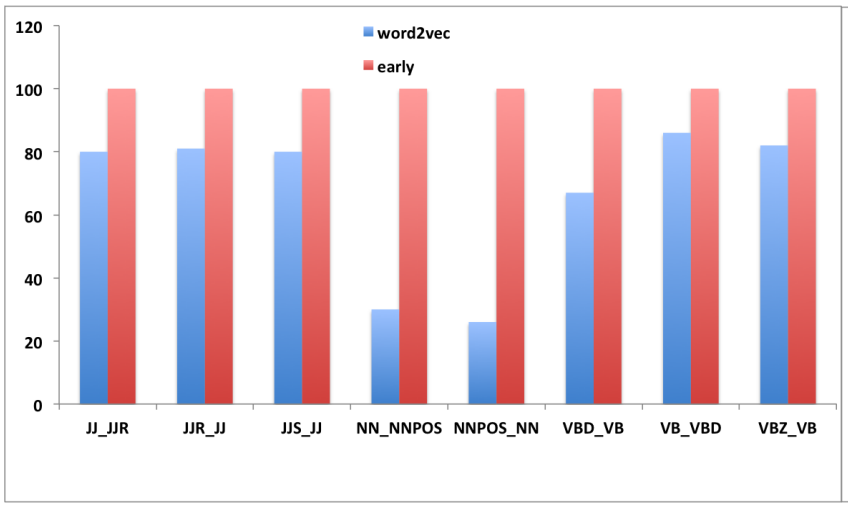


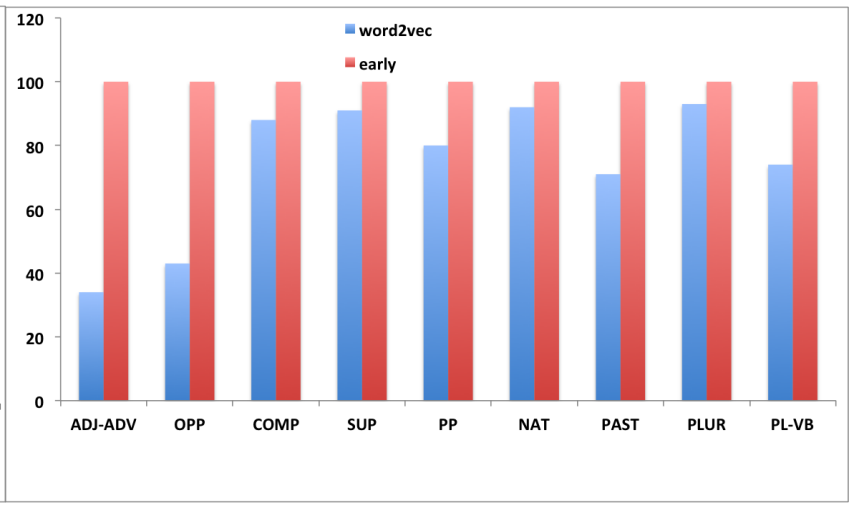
A Comparison to State-of-the-art

- trust `early` if the solution defines with the equation a formal analogy, `word2vec` otherwise

	`msr`	`google`
`word2vec`	67%	63%
`early+word2vec`	72%	71%
Levy et al., 2015	72.9%	75.8%

Comparison by Types





Dataset

- Much harder to get. We did this:
  1. Train a phrasal translation table (from an EN-ES parallel corpus)

```
a actualizar los acuerdos | to update the agreements | 0.0047
a cambiar la base | to change the basis | 0.0035545
basado en el trabajo de | based on the efforts of | 2.02579e-
```

  2. Split good enough associations into R a reference set, and M a translation memory
  3. For each $(s, e) \in R$,
     - use an analogical device to translate s into e, using M as a memory, that is, finding $(x, x'), (y, y'), (z, z') \in M$ such that: $[x \setminus y \bigcirc z \supset s]$ and $[x' \setminus y' \bigcirc z' \supset e]$
     - collecting $[x' \setminus y' \bigcirc z' \supset e]$ (analogies in the English side)
  4. Select:
     - `simple`: a random subset of those equations
     - `hard`: a sampling of equations more difficult for `alea` to solve

Dataset: `simple`

`simple` phrase's avr. length: 16

[	*international investigation*	\	*international democracy*	◯	
	an international investigation	⊃	*an international democracy*		]
[	*young girls*	\	*training of young girls*	◯	
	young girls and	⊃	*training of young girls and*		]
[	*political situation is viable*	\	*political situation is still*	◯	
	the political situation is viable	⊃	*the political situation is still*		]

Dataset: `hard`

`hard` phrase's avr. length: 17

[	*adopted recently by* *study published recently by*	\ ⊃	*recently adopted by* *study recently published by*	◯]
[	*competition and the* *competition and of the*	\ ⊃	*competition and against* *of competition and against*	◯]
[	*their governments to* *their governments and to*	\ ⊃	*their governments are* *and their governments are*	◯]

Accuracy (Silence)

test $\equiv$ `simple` Structured solvers trained on 10 epochs with $\eta = 7$, against `alea` with $\rho = 1000$

train	`simple`	`hard`
`standard`	30.6 (7)	30.1 (10)
`early`	**38.4** (8)	27.6 (10)
`late`	26.9 (9)	20.3 (11)
`safe`	25.4 (8)	25.6 (13)
`alea`	33 (0)	

- `alea` is competitive, but outperformed by `early`
- preferable to train on simple analogies

Accuracy (Silence)

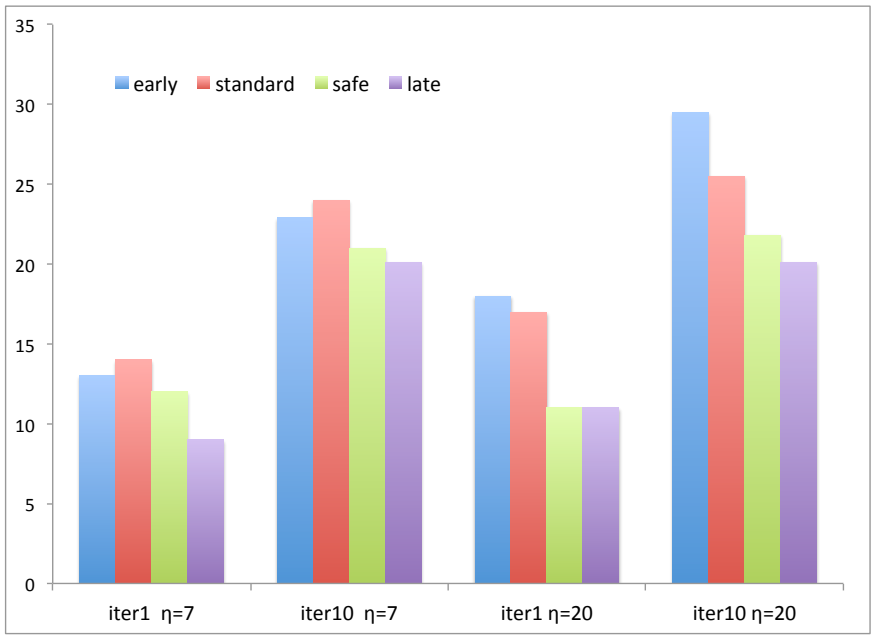
test $\equiv$ `hard` Structured solvers trained on 10 epochs with $\eta = 7$, against `alea` with $\rho = 1000$

train	`simple`	`hard`
`standard`	**19.6** (17)	**24.0** (56)
`early`	**26.0** (6)	**22.9** (7)
`late`	**18.9** (9)	**20.1** (6.7)
`safe`	14.6 (18.7)	**21.0** (56)
`alea`	18 (0)	

- more challenging dataset !
- structured learning systematically better
- preferable to train on harder to solve analogies
- silence rate rather high

Accuracy on `hard`

- after 1 and 10 epochs
- for 2 values of η: 7 and 20



- more epochs is (of course) preferable
- opening the search space as well (expectedly)
- `early` seems overall the best variant tested


Conclusion

Futur

En cours

- probabiliser le solveur (thèse de Rafik Rhouma)

Long terme

- probabiliser tout le processus (search, solveur, agrégation)
- passage à l'échelle
- noyau analogique ?
- analogie sur les arbres ***[?, ?]***

Bibliography I