

Unsupervised Pattern Learning from Sequential Experiences

Anton Kolonin

akolonin@aigents.com

Facebook: akolonin

Telegram: akolonin

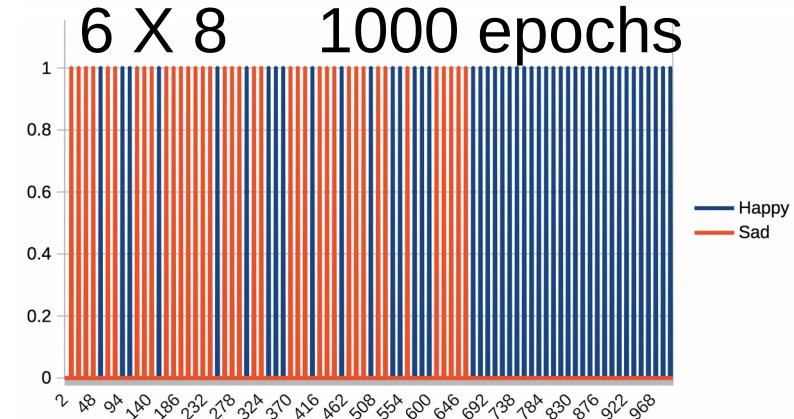
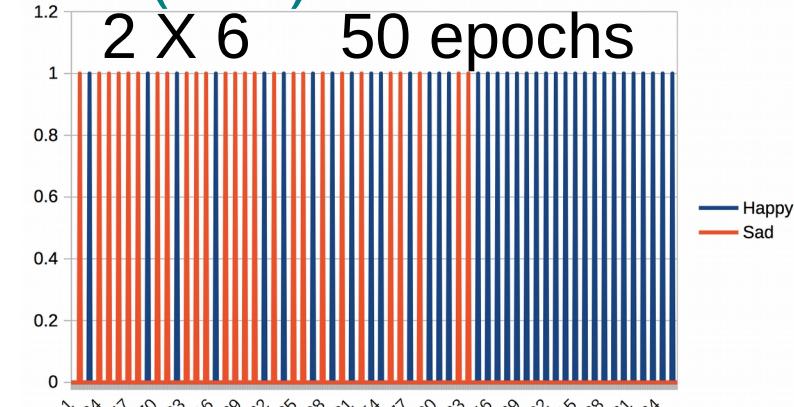
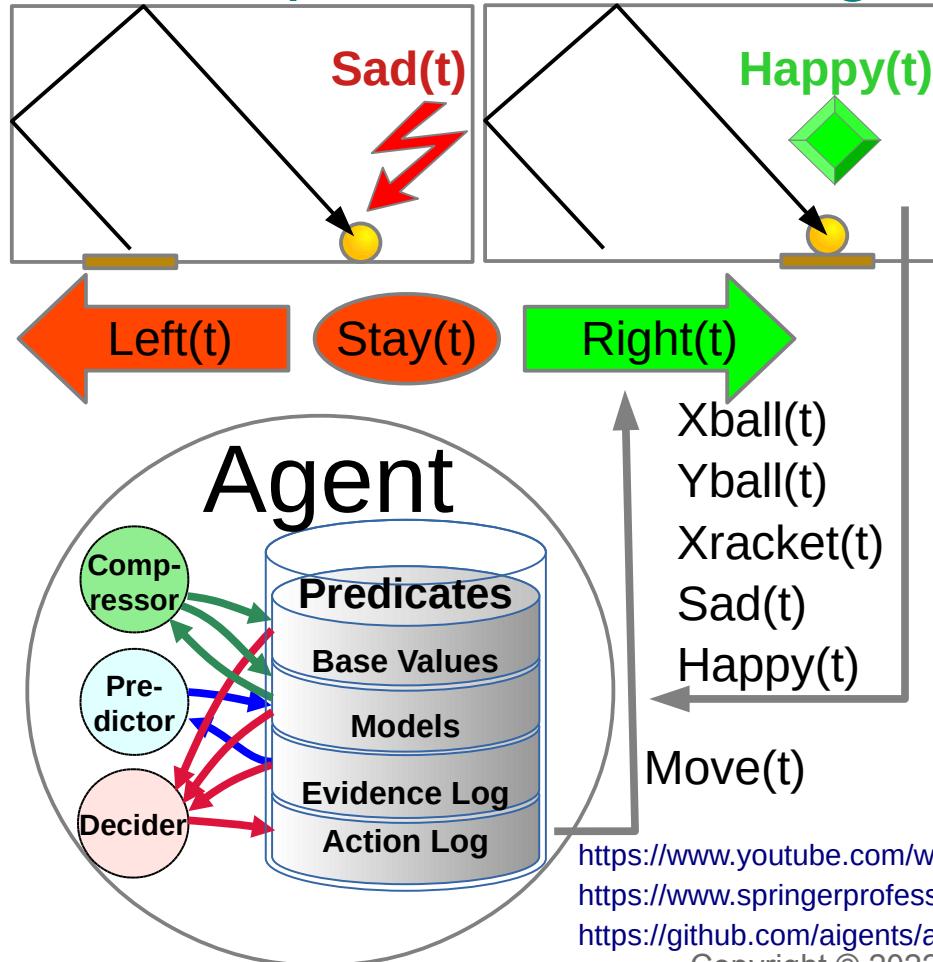


Motivation

Identifying successful/unsuccessful sequential experiences for experiential learning for global (self)reinforcement

Discovering NLP patterns such as words and punctuation for further unsupervised language learning

Identifying successful/unsuccessful sequential experiences for experiential learning with global (self)reinforcement

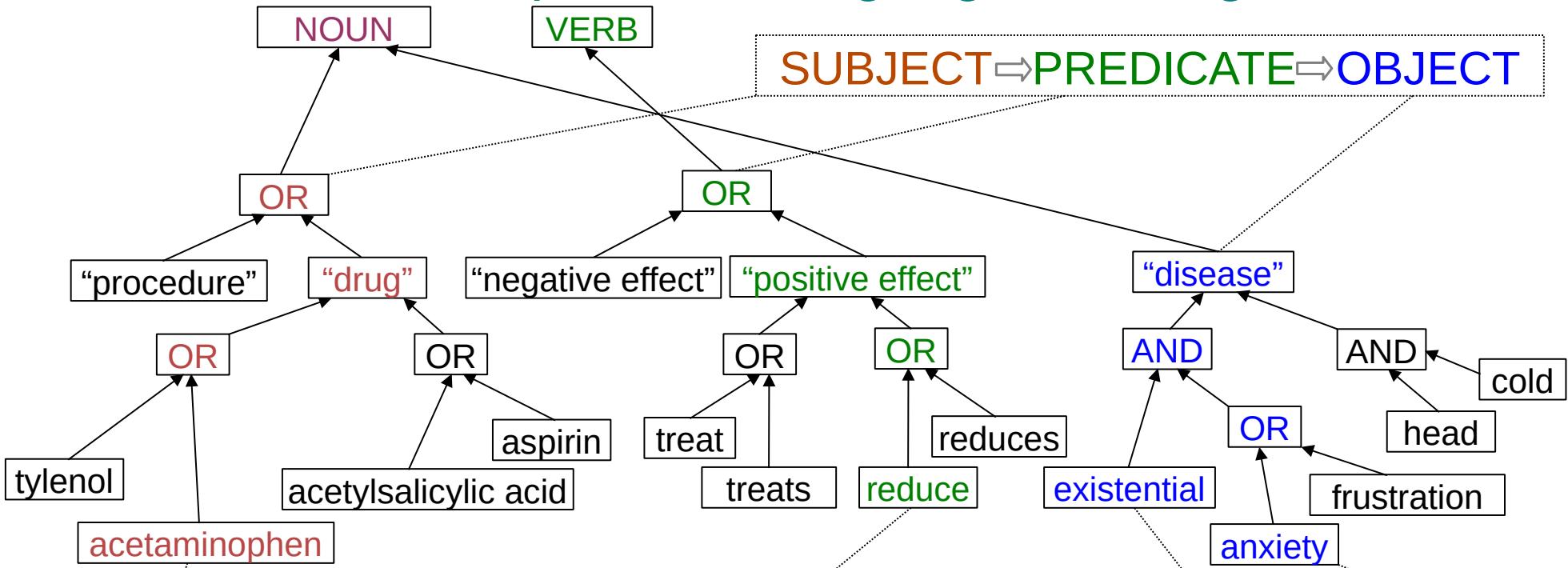


<https://www.youtube.com/watch?v=2LPLhJKh95g>

<https://www.springerprofessional.de/neuro-symbolic-architecture-for-experiential-learning-in-discret/20008336>

<https://github.com/aigents/aigents-java/tree/master/src/main/java/net/webstructor/agj>

Discovering NLP patterns such as words or phrase structures
for unsupervised language learning



acetaminophen → may → significantly → reduce → feelings → of → existential → anxiety

<https://ieeexplore.ieee.org/document/7361868>
<https://github.com/aigents/aigents-jaya>

<https://www.springerprofessional.de/unsupervised-language-learning-in-opencog/15995030>
<https://www.springerprofessional.de/en/programmatic-link-grammar-induction-for-unsupervised-language-le/17020348>
<https://github.com/sinqnet/language-learning/>

Issues to Address

Absence of explicit start/stop tags in continuous streams of spaces in experiential (reinforcement/self-reinforcement) learning with delayed/sparse feedback

<https://www.youtube.com/watch?v=2LPLhJKh95g>

<https://www.springerprofessional.de/neuro-symbolic-architecture-for-experiential-learning-in-discret/20008336>

<https://github.com/aigents/aigents-java/tree/master/src/main/java/net/webstructor/agi>

Complex, cumbersome, unreliable and expensive language-specific tokenization process for unsupervised language learning in NLP

Low quality of unsupervised parsing and tokenization learning based on mutual information and conditional probabilities

<https://www.springerprofessional.de/unsupervised-language-learning-in-opencog/15995030>

<https://www.springerprofessional.de/en/programmatic-link-grammar-induction-for-unsupervised-language-le/17020348>

<https://github.com/singnet/language-learning/>

<https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=6983&context=etd>

Tokenization or Text Segmentation as Language Modeling

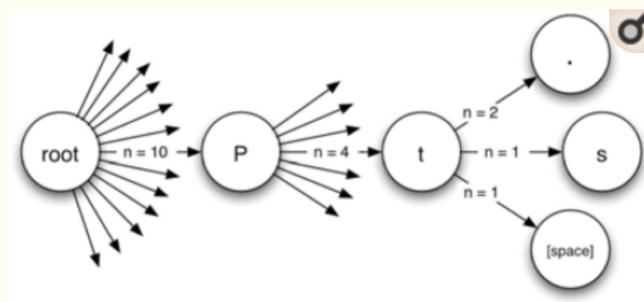


Figure 1

Trie data structure. The probability of observing an ‘s’ given the preceding string “Pt” is $\frac{1}{4}$, or 25%. The freedom following “pt” is 3.

<https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?article=6983&context=etd>
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655800/>

Metrics/Indicators:

Mutual Information

Conditional Probability

Transition Freedom

<https://github.com/aigents/pygents>

Contrastive Evaluation: Test Specific Phenomena

To test if your LM knows something very specific, you can use contrastive examples. These are the examples where you have several versions of the same text which differ only in the aspect you care about: one correct and at least one incorrect. A model has to assign higher scores (probabilities) to the correct version.

The roses in the vase by the door ?

Competing answers: **is, are**

P(The roses in the vase by the door **are**)

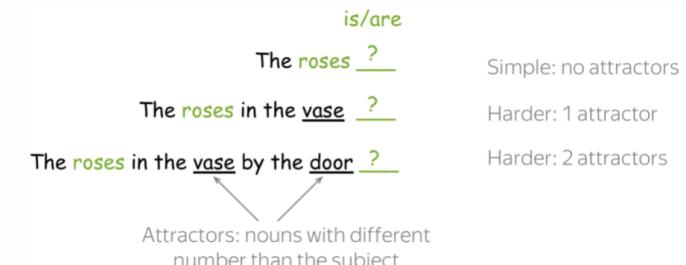
Is the correct answer ranked higher?

P(The roses in the vase by the door **is**)

$P(\dots\text{are}) > P(\dots\text{is})$?

A very popular phenomenon to look at is subject-verb agreement, initially proposed in the [Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies](#) paper. In this task, contrastive examples consist of two sentences: one where the verb agrees in number with the subject, and another with the same verb, but incorrect inflection.

Examples can be of different complexity depending on the number of attractors: other nouns in a sentence that have different grammatical number and can “distract” a model from the subject.



https://lena-voita.github.io/nlp_course/language_modeling.html

Claims

Transition Freedom (TF) appears to be superior (over **Mutual Information** and **Conditional Probability**) for unsupervised text segmentation (tokenization).

English and Russian require one specific way (variance) of handling the TF while Chinese requires a bit different specific way (derivative) for the same purpose.

Tokenization quality for Russian and English may be as high as $F1=0.96-1.0$, depending on training and testing corpora while for Chinese the minimum is $F1=0.83-0.92$, depending on the assessment assumptions.

Larger training corpora does non necessarily effect in better tokenization quality, while compacting the models eliminating statistically weak evidence typically improve the quality.

TF-based tokenization appear quality same or better than lexicon-based one for Russian and English while for Chinese appears the opposite (as it could be anticipated).

Doing Russian and English tokenization with removed spaces makes the situation similar to Chinese with reasonable quality on lexicon-based tokenization but much worse results on TF-based one.

<https://github.com/aigents/pygents>

Corpora and Methodology

Train corpora

Chinese

CLUE News 2016 Validation – 270M

CLUE News 2016 Train – 8,500M

English

Brown – 6M

Gutenberg Children – 29M

Gutenberg Adult – 140M

Social Media – 68M

All above – combined

Russian

RusAge Test – 141M

RusAge Previews – 825M

Test corpus

Parallel Chinese/English/Russian

– 100 multi-sentence statements on finance

Metrics/Indicators:

Ngram (Character)

Probability or Conditional Transition Probability (p-/p+)

Deviation (dvp-/dvp+)

Derivative (dp-/dp+)

Transition Freedom (f-/f+)

Deviation (dvf-/dvf+)

Derivative (df-/df+)

Hyper-parameters:

Combination of Ngram ranks N ([1],[2],[3],[1,2],[1,2,3],...)

Threshold for model compression

Threshold for segmentation

Evaluations:

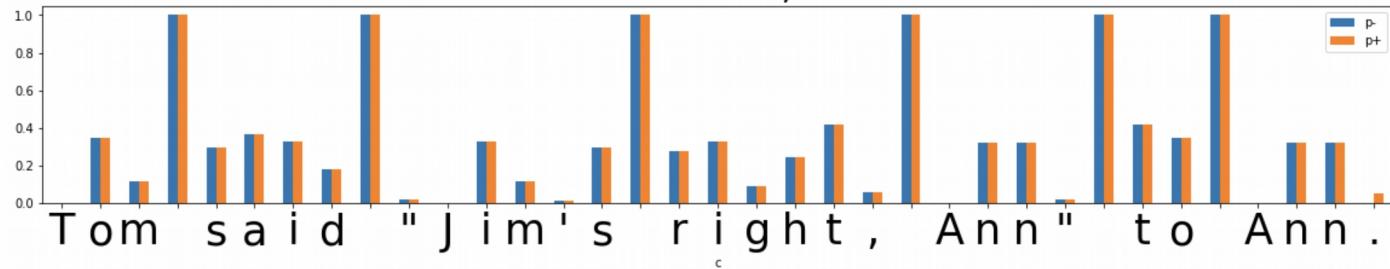
Tokenization F1, on set of tokens found comparing to delimiter-based (English/Russian) or Jieba (Chinese)

Precision on set of tokens found comparing to reference lexicons

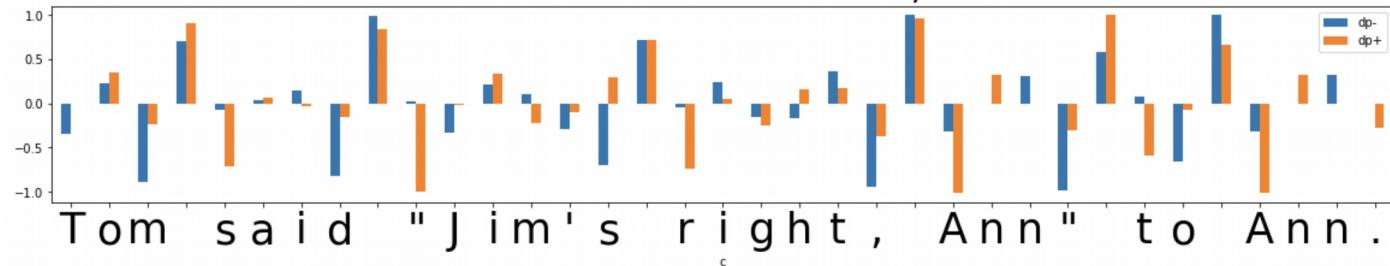
Unsupervised Text Segmentation (Tokenization)

Metrics/Indicators:
Ngram (Character)
Probability

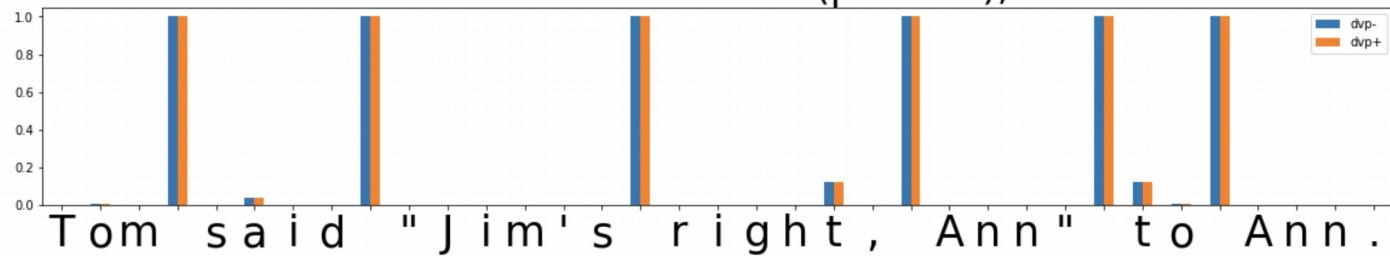
Probabilities, N=1



Probabilities - derivatives, N=1



Probabilities - deviations (positive), N=1



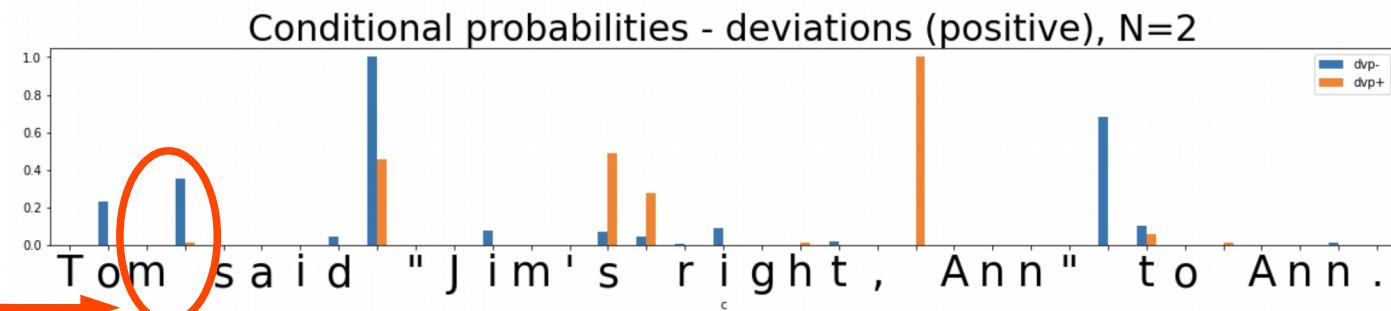
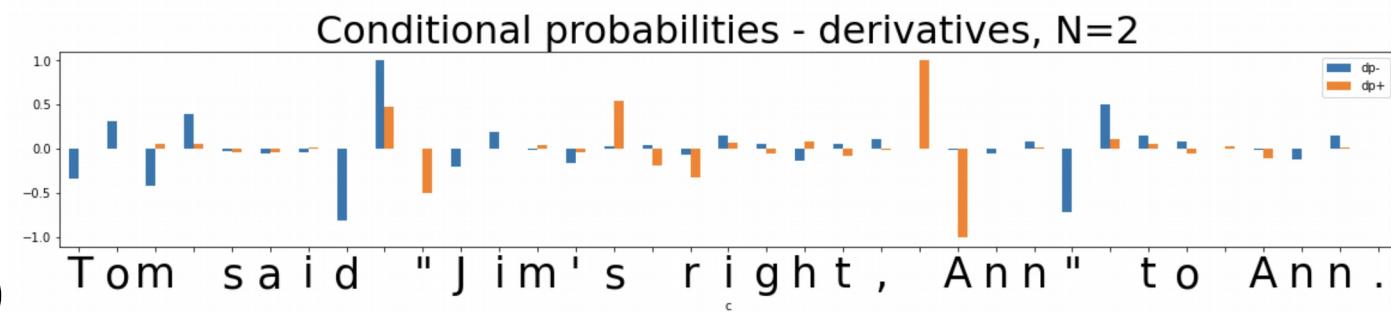
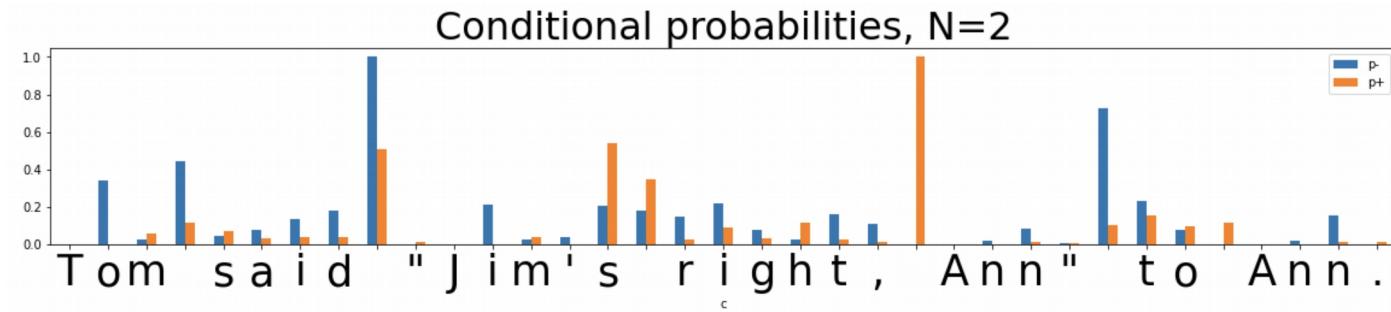
Unsupervised Text Segmentation (Tokenization)

Metrics/Indicators:

Ngram (Character)
Conditional
Probability
(of Transition)

$P(\text{Ngram}_{n+1})/P(\text{Ngram}_n)$

$P("m")/P(m")$



Unsupervised Text Segmentation (Tokenization)

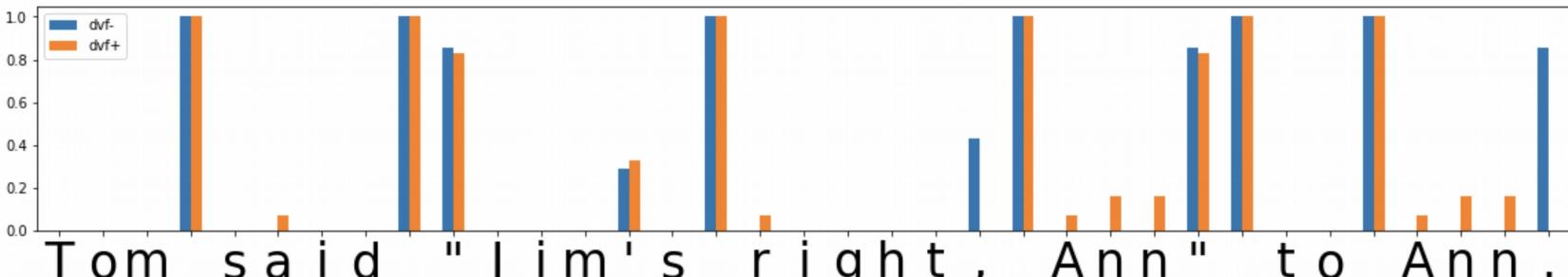
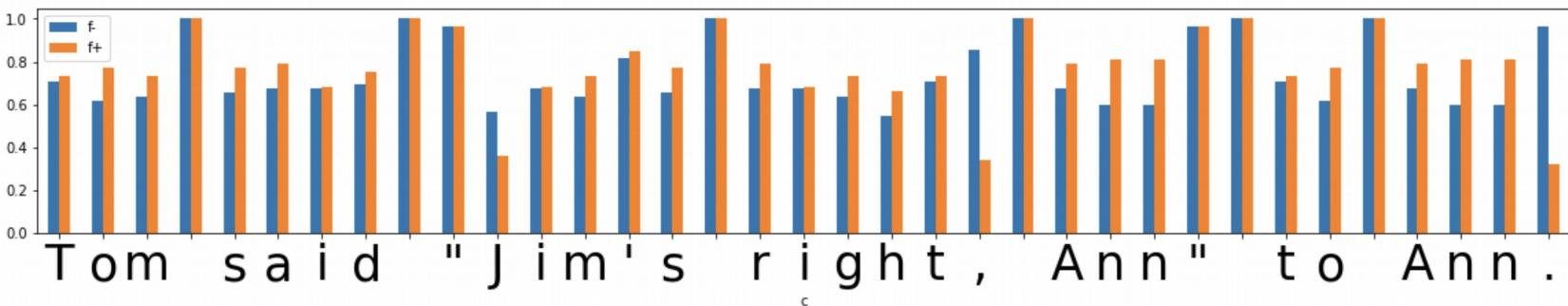
Threshold 0.25
Tom said "Jim's right, Ann" to Ann.
['Tom', ' ', 'said', ' ', ' ', "Jim's", ' ', 'right', ' ', ' ', 'Ann', ' ', ' ', 'to', ' ', ' ', 'Ann', '.']
['Tom', ' ', 'said', ' ', ' ', "Jim", " ", "s", ' ', 'right', ' ', ' ', 'Ann', ' ', ' ', 'to', ' ', ' ', 'Ann', '.']
0.89

Threshold 0.35
Tom said "Jim's right, Ann" to Ann.
['Tom', ' ', 'said', ' ', ' ', "Jim's", ' ', 'right', ' ', ' ', 'Ann', ' ', ' ', 'to', ' ', ' ', 'Ann', '.']
['Tom', ' ', 'said', ' ', ' ', "Jim's", ' ', 'right', ' ', ' ', 'Ann', ' ', ' ', 'to', ' ', ' ', 'Ann', '.']
1.0

Metrics/ Indicators:

Transition
Freedom
Deviation

(Freedom
of Transition)



Unsupervised Text Segmentation (Tokenization)

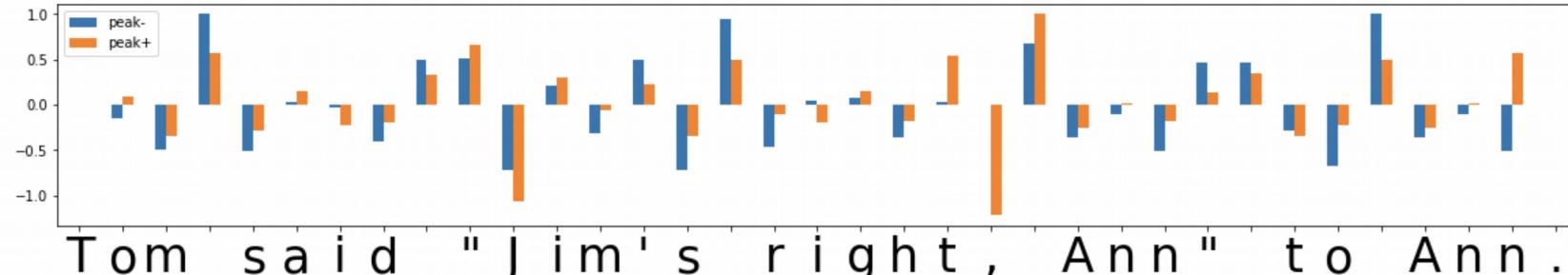
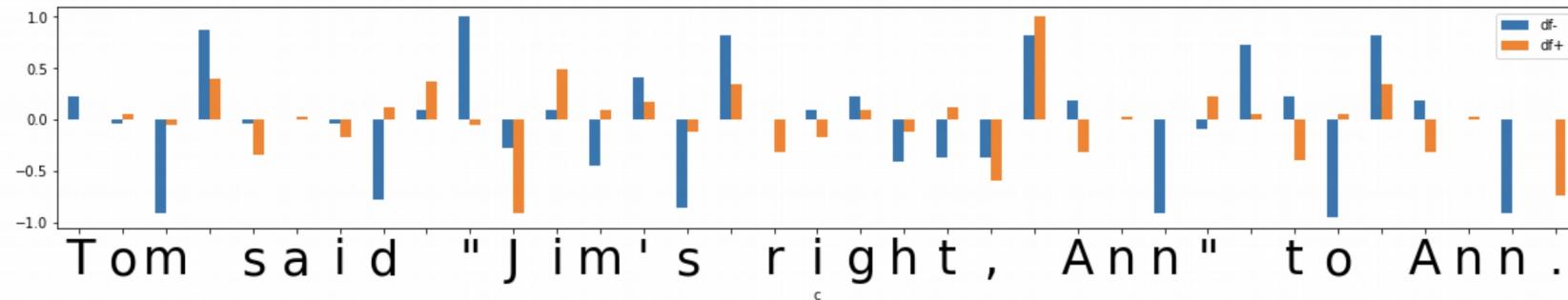
Metrics/

Indicators:

Transition
Freedom
Derivative
and “Peak”

(Freedom
of Transition
“Peak”)

Threshold 0.25
Tom said "Jim's right, Ann" to Ann.
['Tom', ' ', 'said', ' ', "'", "Jim's", ' ', 'right', ' ', ' ', 'Ann', ' ", " ', 'to', ' ', 'Ann', '.']
['Tom', ' ', 'said', ' ', "'", 'Ji', 'm', 's', ' ', 'right', ' ', ' ', 'Ann', ' ", " ', 'to', ' ', 'Ann', '.']
0.89
Threshold 0.35
Tom said "Jim's right, Ann" to Ann.
['Tom', ' ', 'said', ' ', "'", "Jim's", ' ', 'right', ' ', ' ', 'Ann', ' ", " ', 'to', ' ', 'Ann', '.']
['Tom', ' ', 'said', ' ', "'", 'Jim', 's', ' ', 'right', ' ', ' ', 'Ann', ' ", " ', 'to', ' ', 'Ann', '.']
0.82



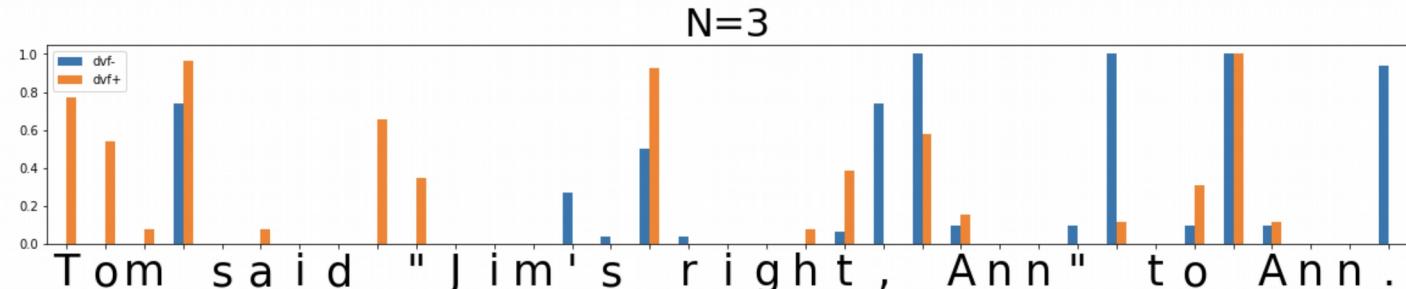
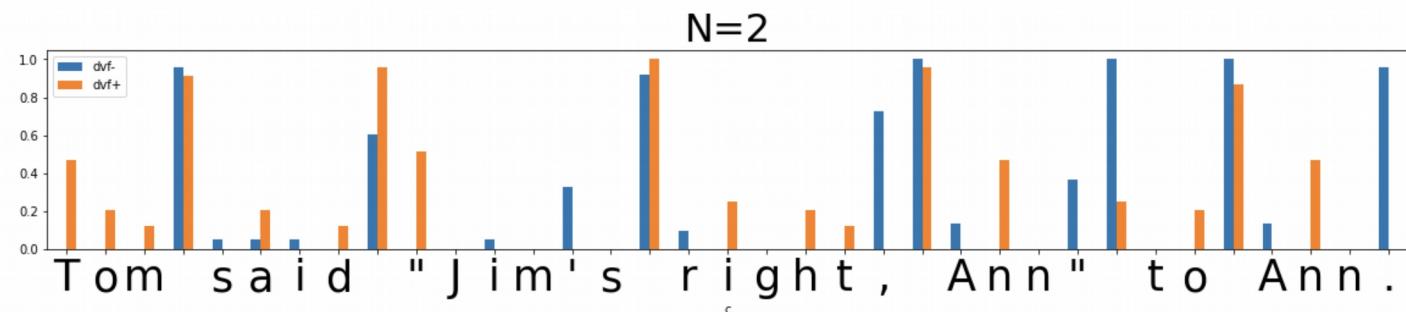
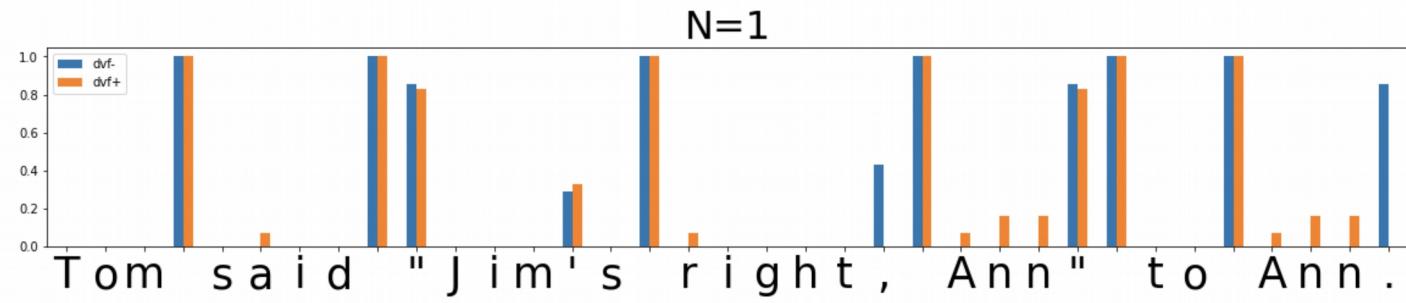
Unsupervised Text Segmentation (Tokenization)

Metrics/

Indicators:

Transition
Freedom
Deviation

(varying “N”)



Unsupervised Text Segmentation (Tokenization)

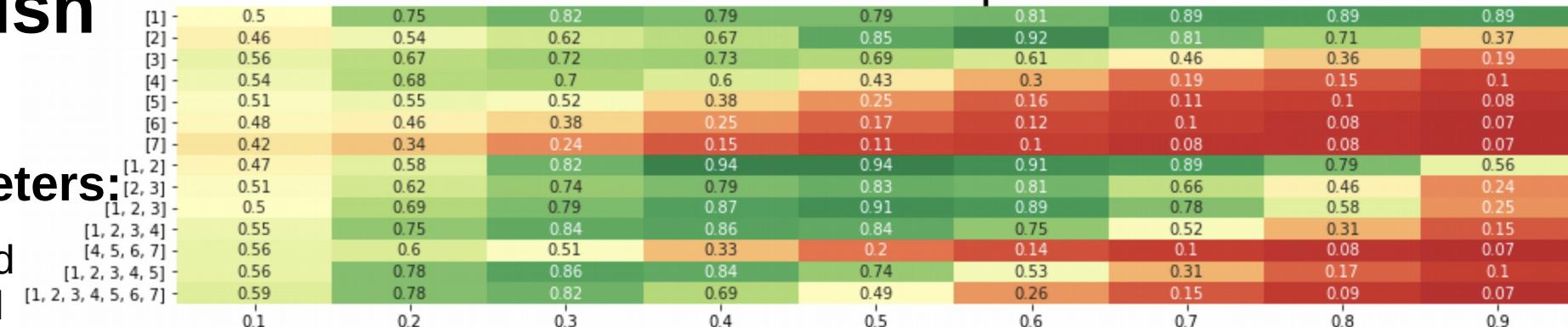
English

Hyper-

Parameters:

Threshold
for model
compression

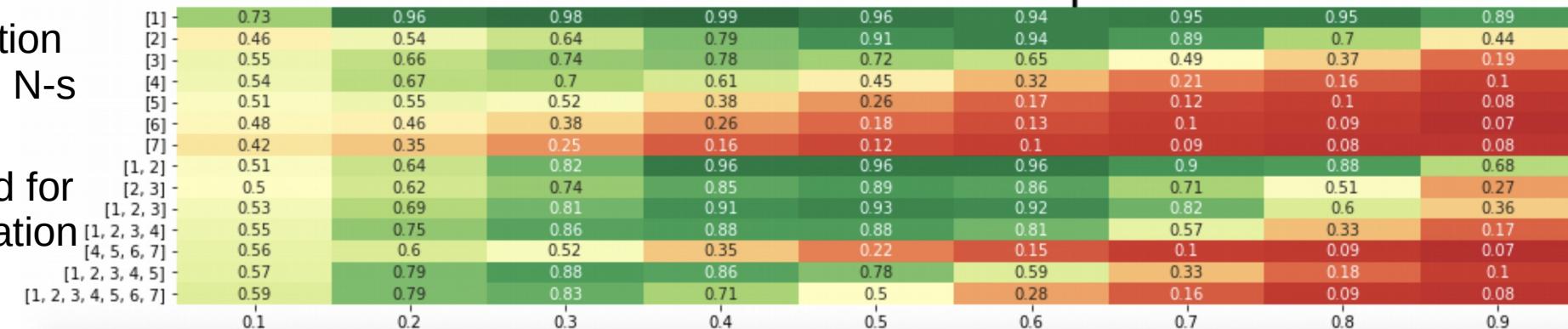
F1 - Brown ddf- & ddf+ filter=0 parameters=10967135



F1 - Brown ddf- & ddf+ filter=0.0001 parameters=8643703

Combination
of Ngram N-s

Threshold for
segmentation



Unsupervised Text Segmentation (Tokenization)

Chinese

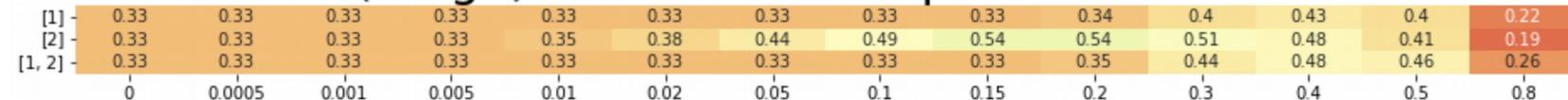
Hyper-
Parameters:

Threshold
for model
compression

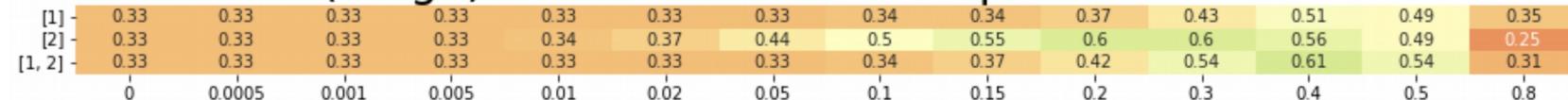
Combination
of Ngram N-s

Threshold for
segmentation

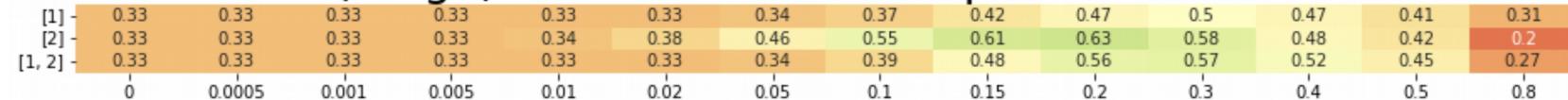
F1 - Train(Large) f- & f+ filter=0 parameters=249859247



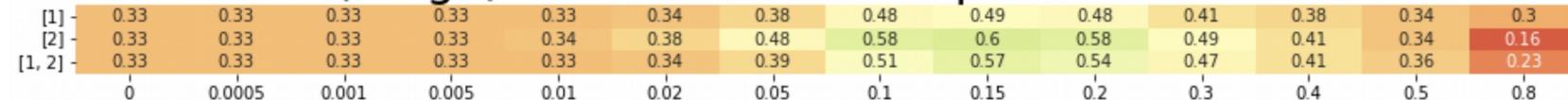
F1 - Train(Large) f- & f+ filter=0.0001 parameters=231751412



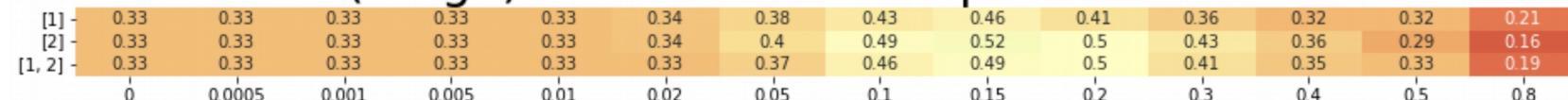
F1 - Train(Large) f- & f+ filter=0.001 parameters=196866127



F1 - Train(Large) f- & f+ filter=0.01 parameters=123792046



F1 - Train(Large) f- & f+ filter=0.1 parameters=51791264



Unsupervised Text Segmentation (Tokenization)

The father told the mother that the child was right.

Threshold 0.15

父亲告诉母亲，孩子是对的。

['父亲', '告诉', '母亲', '，', '孩子', '是', '对', '的', '。']

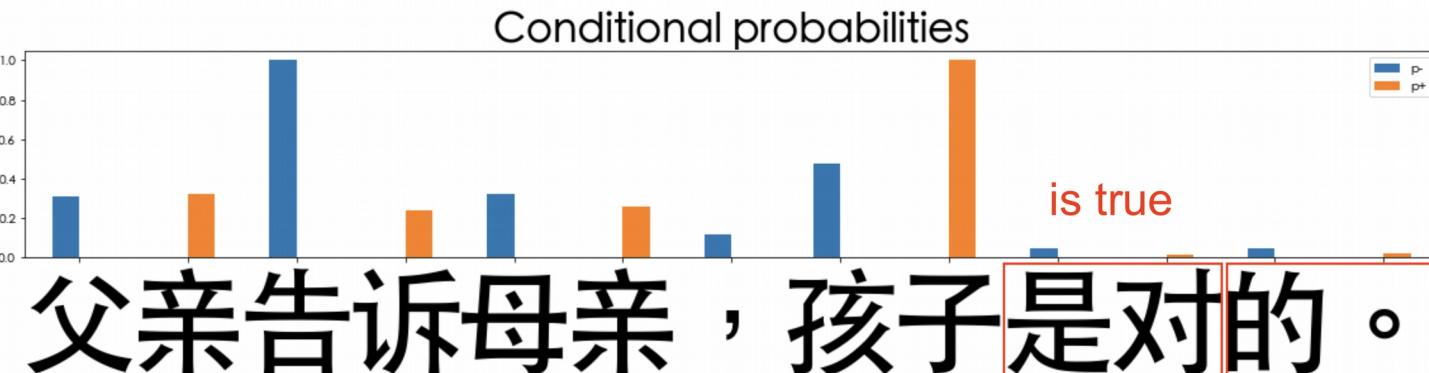
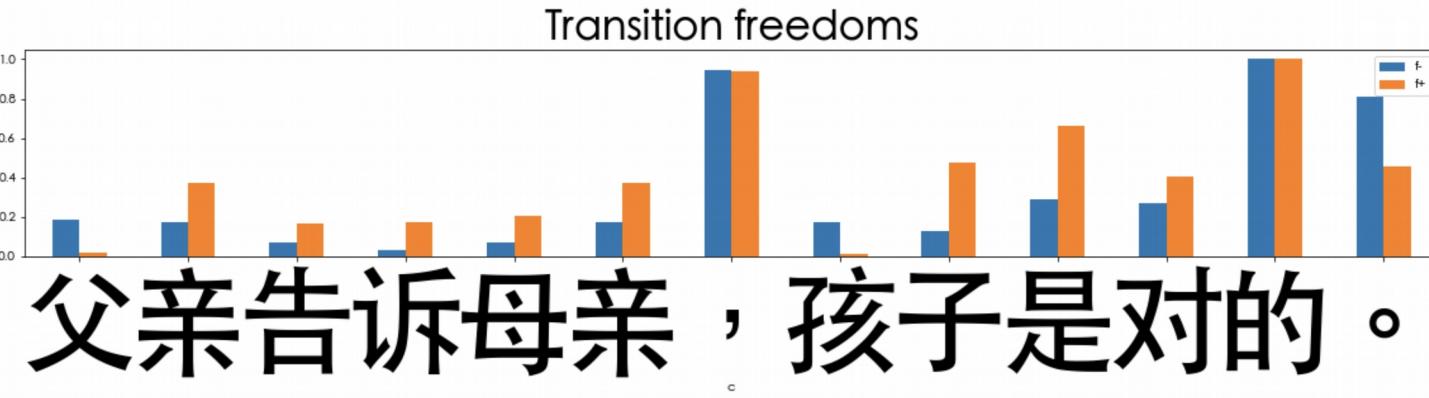
['父亲', '告诉', '母亲', '，', '孩子', '是', '对', '的', '。']

1.0

Metrics/Indicators:

Transition
Freedom
Deviation

Conditional
Probability
(of Transition)

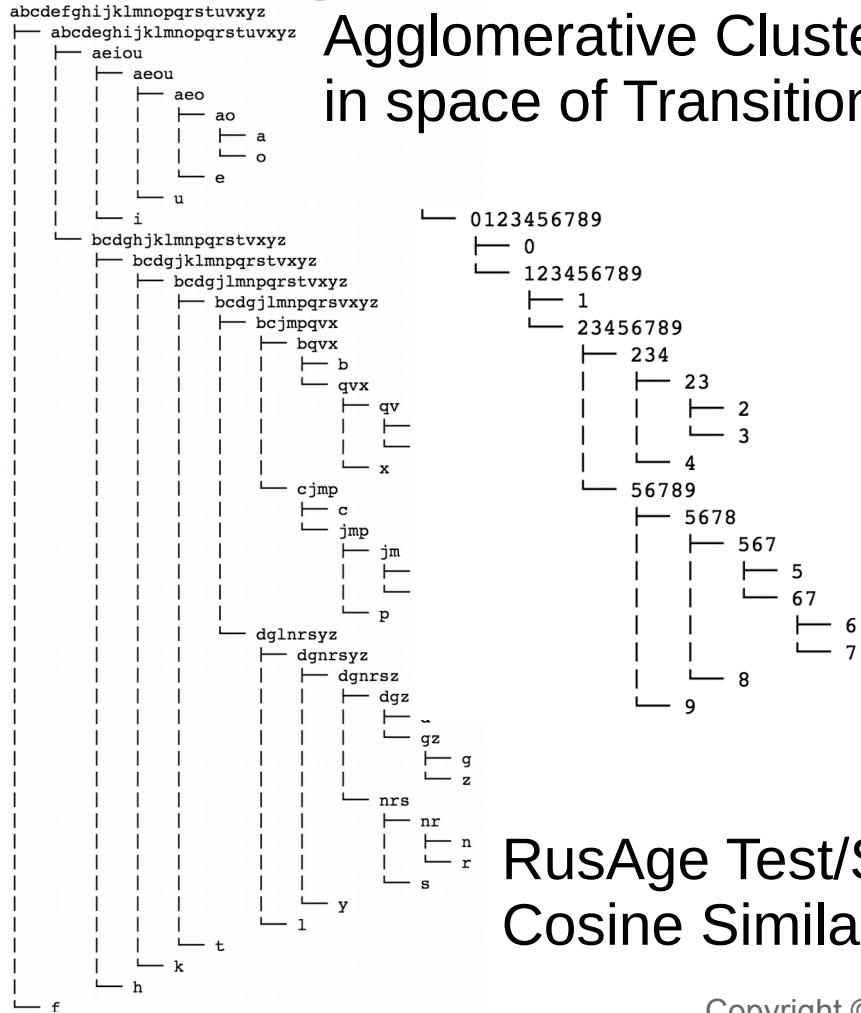


Results – Freedom-based Tokenization against Lexicon

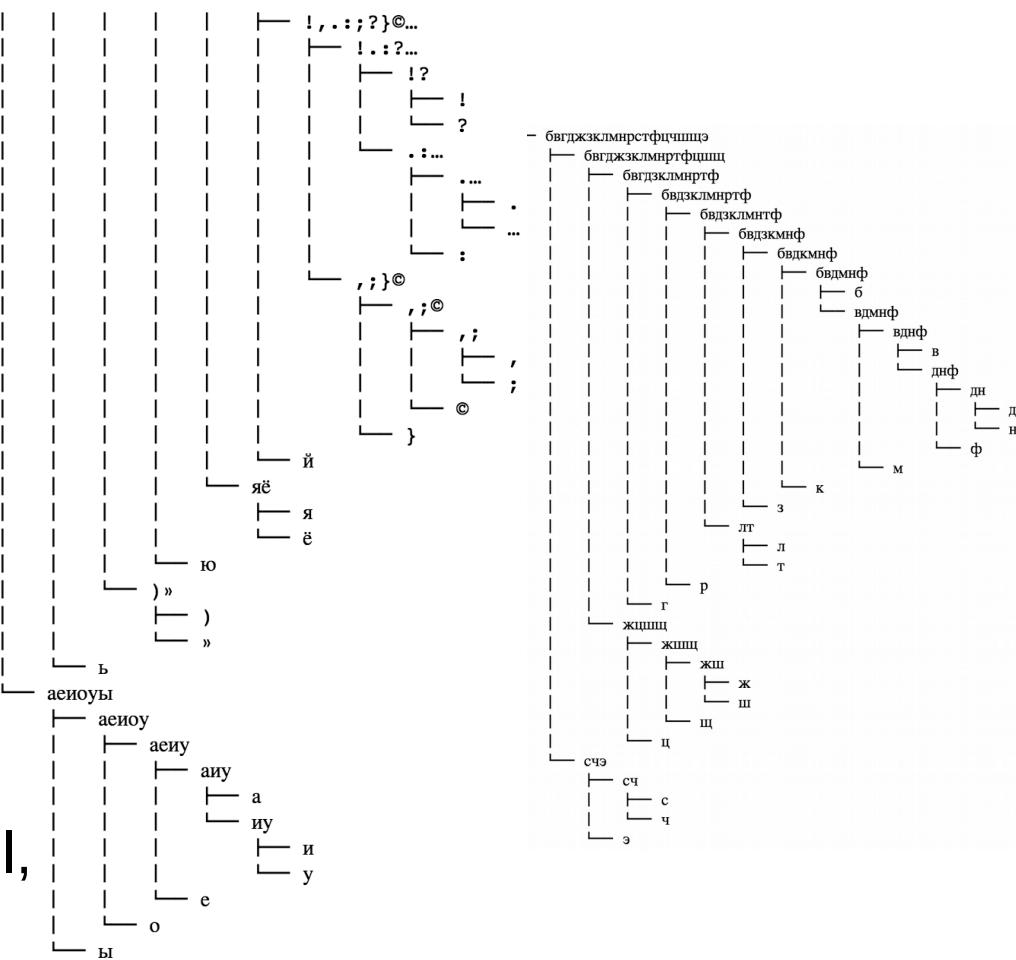
Language	Tokenizer	Tokenization F1	Lexicon Discovery Precision
English	Freedom-based	0.99	0.99 (vs 1.0)
English	Lexicon-based	0.99	-
English no spaces	Freedom-based	0.42	-
English no spaces	Lexicon-based	0.79	-
Russian	Freedom-based	1.0	1.0 (vs 1.0)
Russian	Lexicon-based	0.94	-
Russian no spaces	Freedom-based	0.26	-
Russian no spaces	Lexicon-based	0.72	-
Chinese	Freedom-based	0.71	0.92 (vs 0.94)
Chinese	Lexicon-based	0.83	-

Unsupervised Character Category Learning

Agglomerative Clustering in space of Transitions

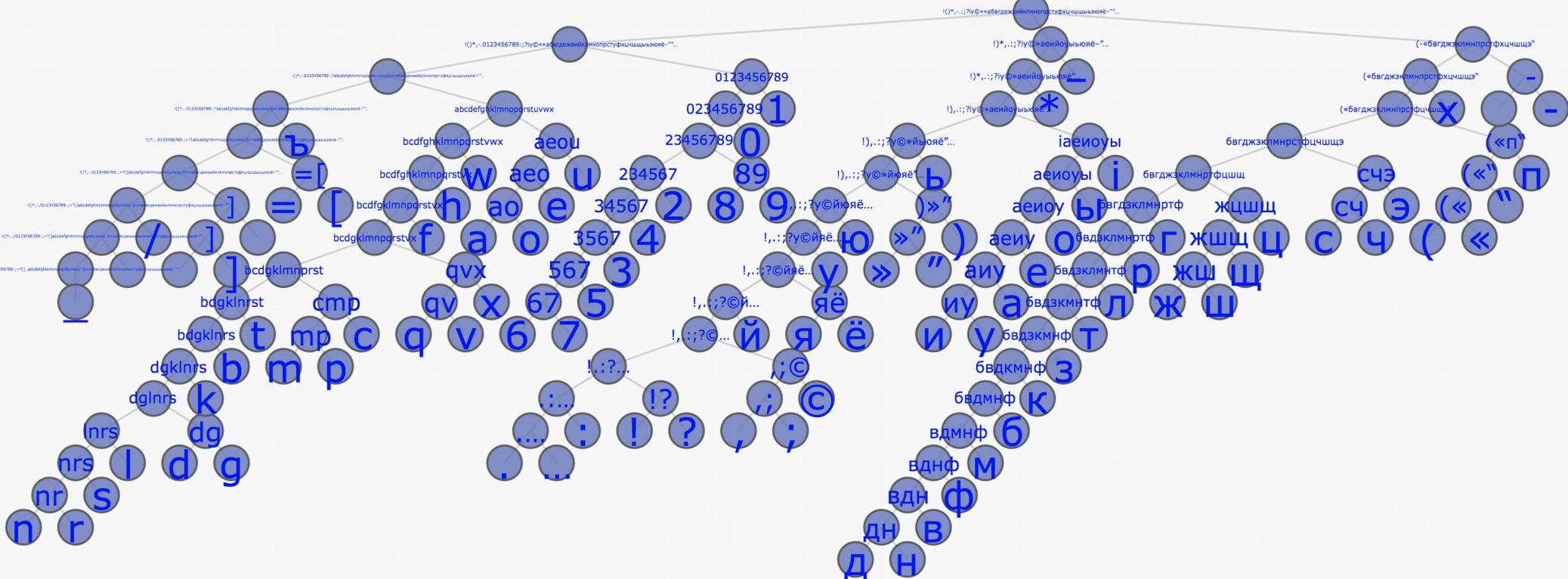


RusAge Test/Small, Cosine Similarity



Unsupervised Character Category Learning

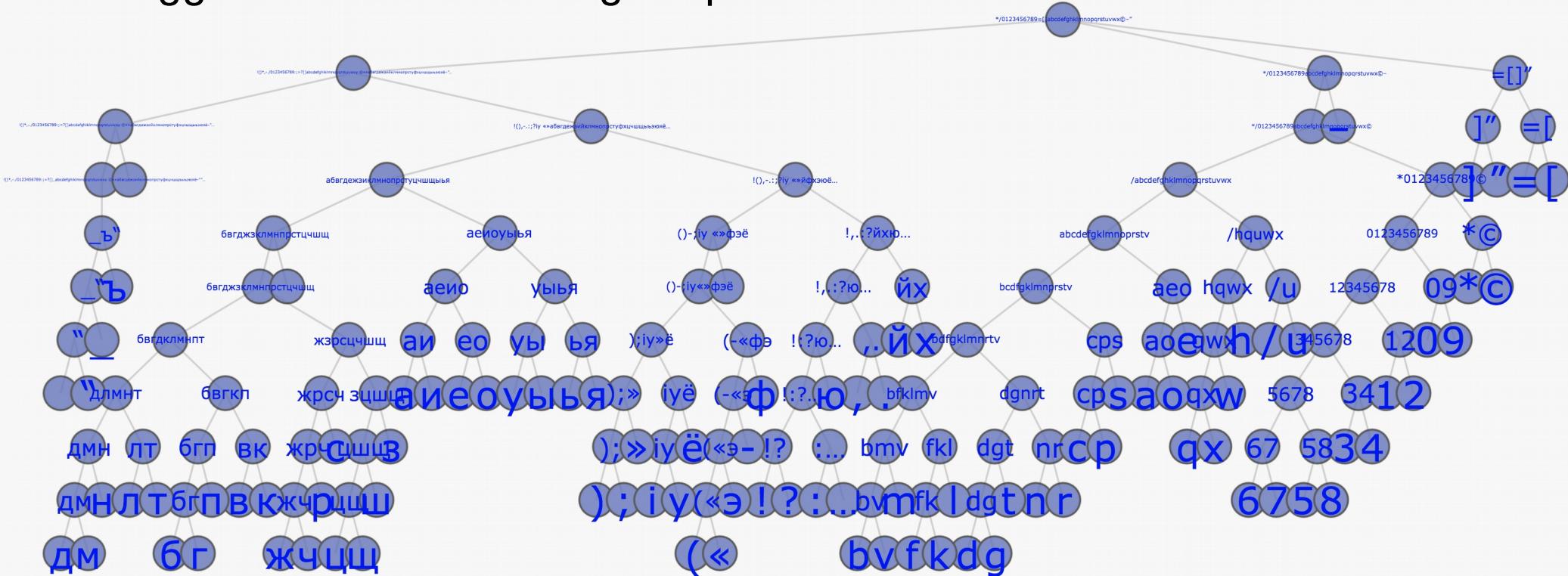
Agglomerative Clustering in space of Transitions



RusAge Previews/Full, Cosine Similarity

Unsupervised Character Category Learning

Agglomerative Clustering in space of Transitions



RusAge Previews/Full, Jackard Similarity

Conclusion and Further Work

Unsupervised Tokenization based on Transition Freedom (TF) recall and precision appears good enough as initial approximation for further application of and self-reinforcement learning as part of experiential learning.

Optimal thresholds and specific TF-based metrics are specific to language and the process and policy of their discovery and adjustment should be further explored.

Clustering or parts of speech on space of transition graphs may provide some insights on morphology and punctuation structure of low-resource and domain-specific languages.

Hybridization of TF-based tokenization approach with lexicon-based one might be efficient for low-resource and domain-specific languages.

Further unsupervised grammar learning experiments can be run on the basis of suggested unsupervised tokenization approach.

Applications for other Experiential Learning environments, including the ones with delayed/sparse feedback.

Using Reinforcement Learning techniques with self-reinforcement on historical data under Unsupervised Learning setup.

<https://github.com/aigents/pygents>

Applications other than NLP and Experiential/Reinforcement Learning

Everything is a
Process of a
Scenario!

<https://arxiv.org/abs/1807.02072>

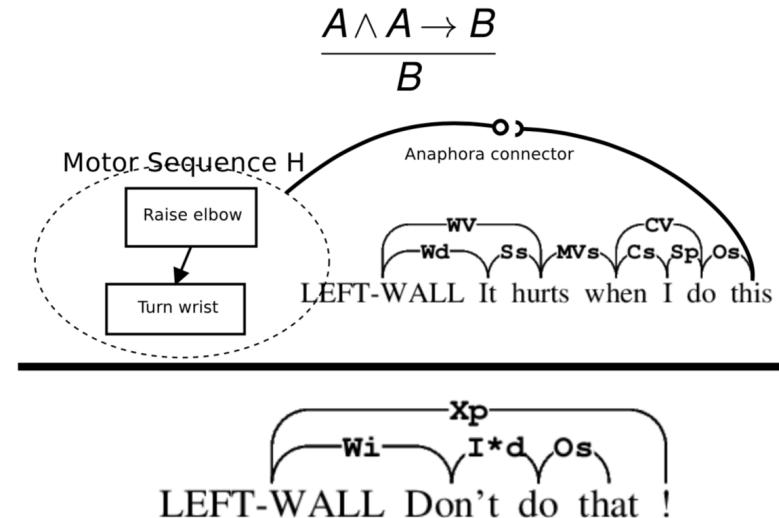
	Spoken language recognition	Written language recognition	Identification of patterns in the text	Causal analysis
Actor	Coefficient on spectrogram for particular frequency	Property of specific stroke: On the top, at the bottom, long, short, skewed, etc.	Object property value: Name “John”	Specific actor: John Doe
Event	Combination of coefficients on spectrogram	Period or stroke composing the letter: .	Object class instance: Name “John”, surname “Doe”	Specific event: John Doe cleaning the window on the second floor.
Coincidence	Specific sound	Coincidence: i	Co-occurrence of object of class person properties: “John Doe”	Specific coincidence: Window on the second floor is dirty, John Doe is cleaning it.
Process	Specific spoken word	Specific written word: ping	Specific phrase: “John Doe cleans the window”	Specific process: Window on the second floor was dirty, John Doe has cleaned it and now it is clean.
Role	Pitch frequency	Property of symbol: orientation, extent, symmetry, etc.	Domain of the object class property: person’s surname	Typical role: Cleaner
Appearance	Spectral cluster on the spectrogram	Element of symbol: .	Class of the variable object: person	Typical appearance: Someone cleaning something
Situation	Sound of speech	Symbol: i	Pattern variable: \$subject	Typical situation: Someone is cleaning something which is dirty.
Scenario	Spoken word accordingly to the language model	Written word: ping	Phrase pattern: “\$subject cleans \$object”	Typical scenario: Something was dirty, someone has cleaned it, it is clean now.

Text Grounding on Non-Text Events

Common Sense Reasoning

Everything is a
Graph and can
be coded with
Link Grammar!

Rules, laws, axioms of reasoning and inference can be learned.



Naively, simplistically: Learned Stimulus-Response AI (SRAI)⁹

Linas Vepstas, 2021

<https://aigents.github.io/inlp/2021/slides/>

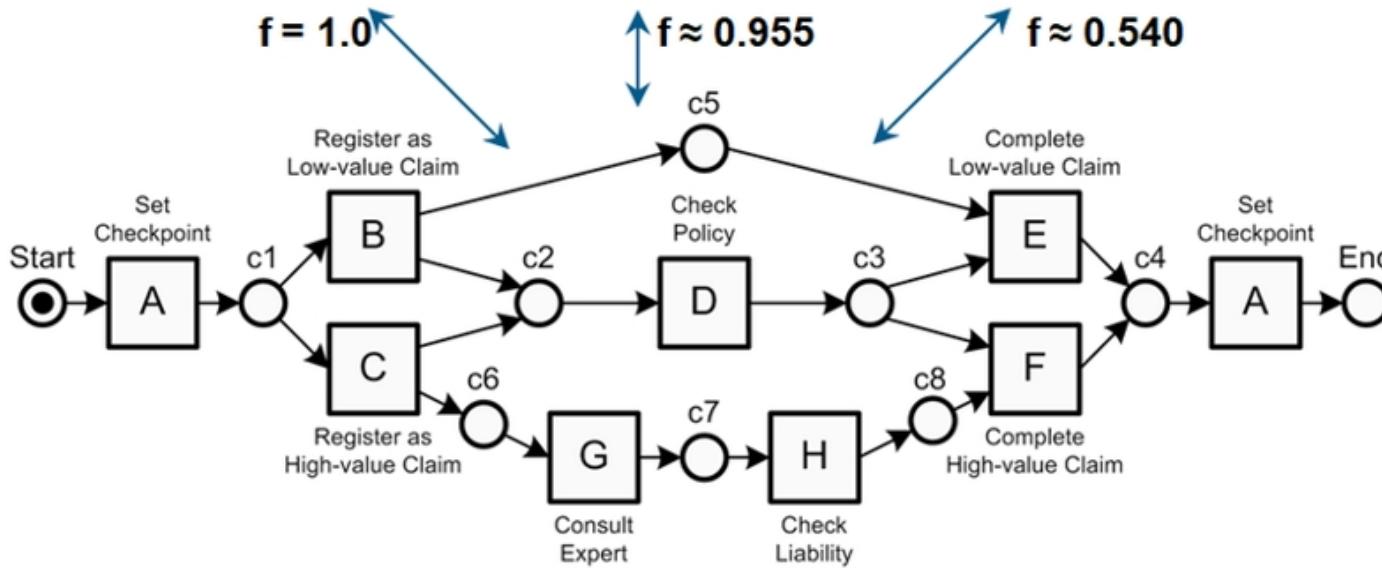
⁹Metaphorical example: Mel'čuk's Meaning Text Theory (MTT) SemR + Lexical Functions (LF) would be better.

Process Mining in Business Log Data

No. of Instances	Log Traces
4070	ABDEA
245	ACDHFA
56	ACGDHFA

No. of Instances	Log Traces
1207	ABDEA
145	ACDHFA
56	ACGDHFA
23	ACHDFA
28	ACDHFA

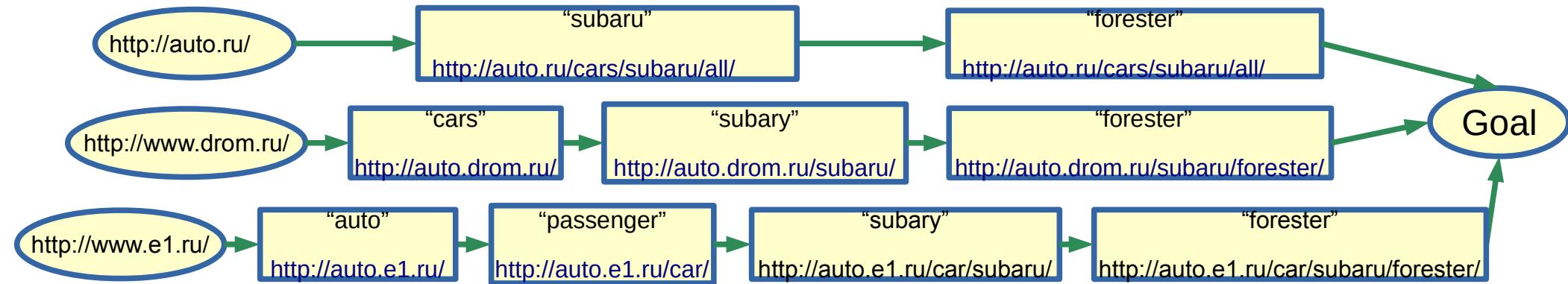
No. of Instances	Log Traces
24	BDE
7	AABHF
15	CHF
6	ADBE
1	ACBGDFAA
8	ABEDA



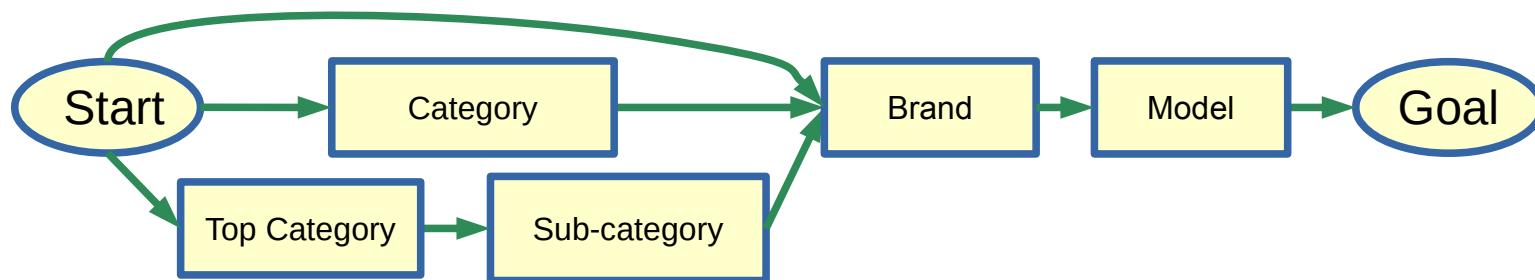
https://www.puzzledata.com/process-mining_eng/

Scenario Mining in Process Data Example

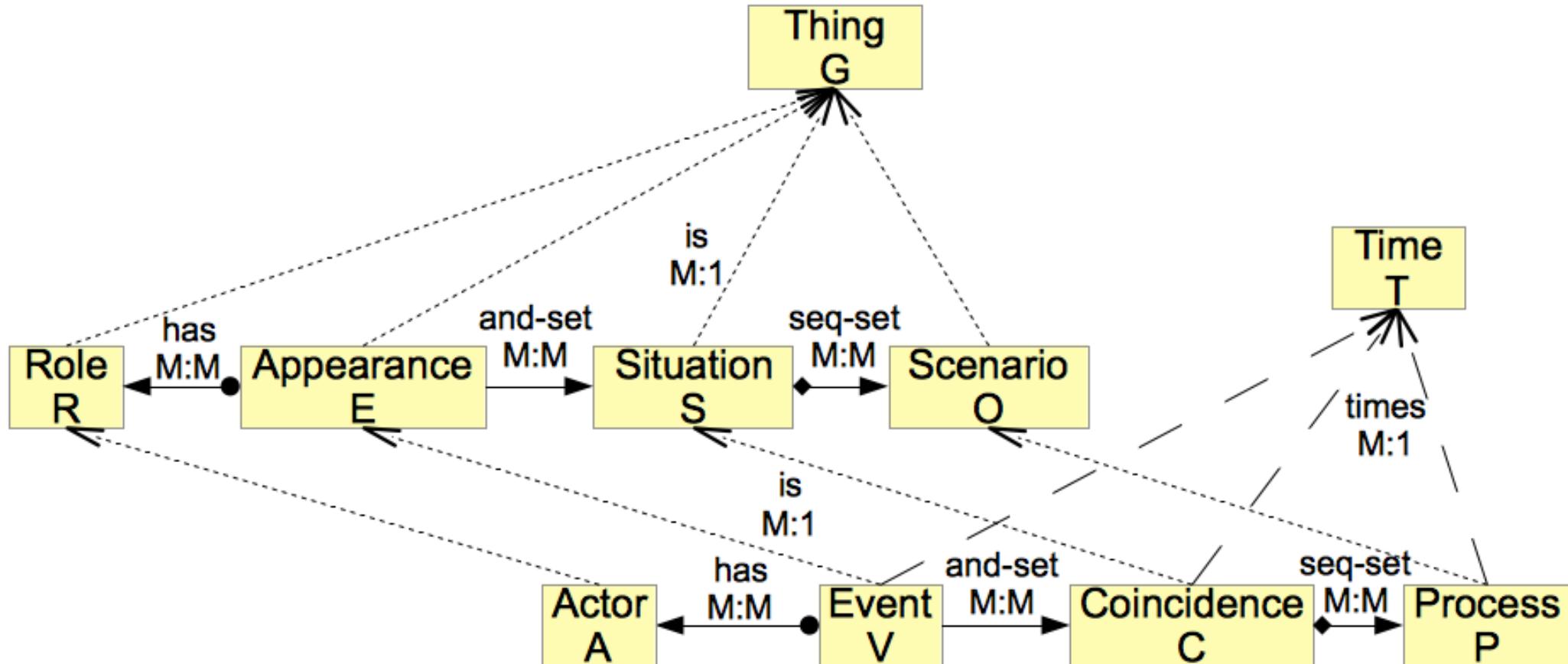
Internet navigation routes when searching for “Forester” cars



Generalized click-through scenario when searching for products

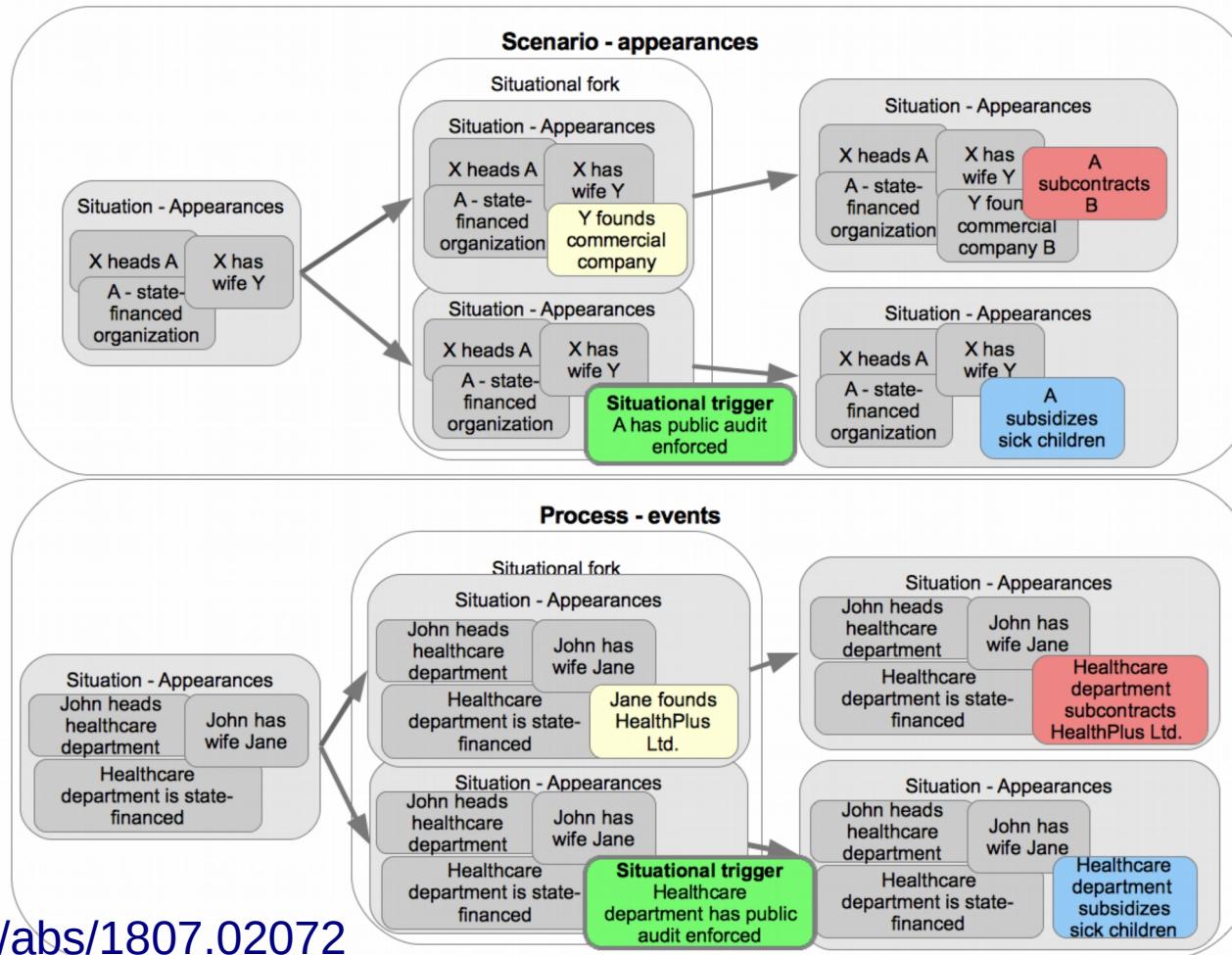


Foundation (Generic) Activity Ontology



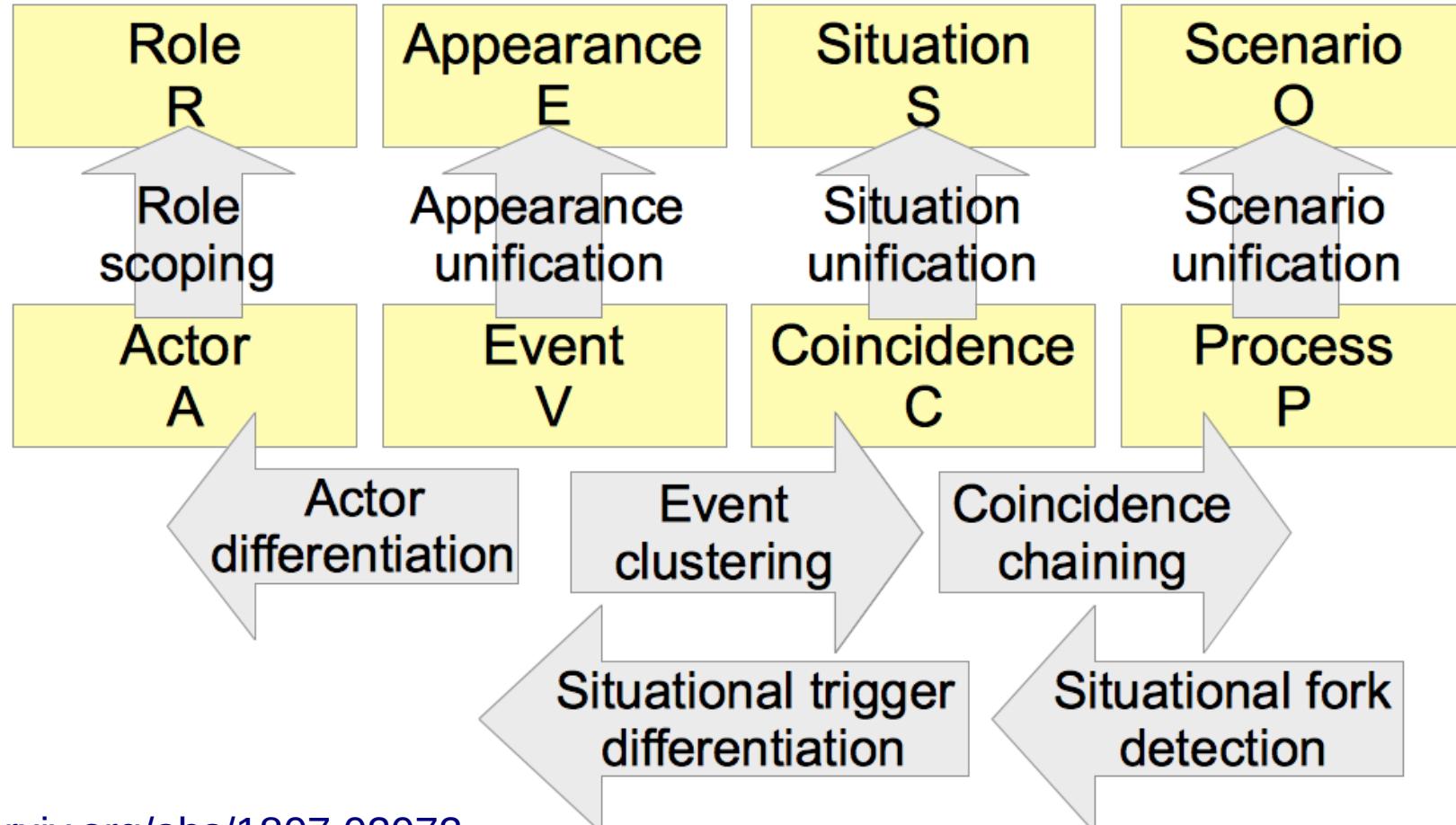
<https://arxiv.org/abs/1807.02072>

Upper (Domain) Activity Ontology Example



<https://arxiv.org/abs/1807.02072>

Scenario Mining in Process Data



<https://arxiv.org/abs/1807.02072>

Thank you and welcome!

Anton Kolonin

akolonin@aigents.com

Facebook: akolonin

Telegram: akolonin

