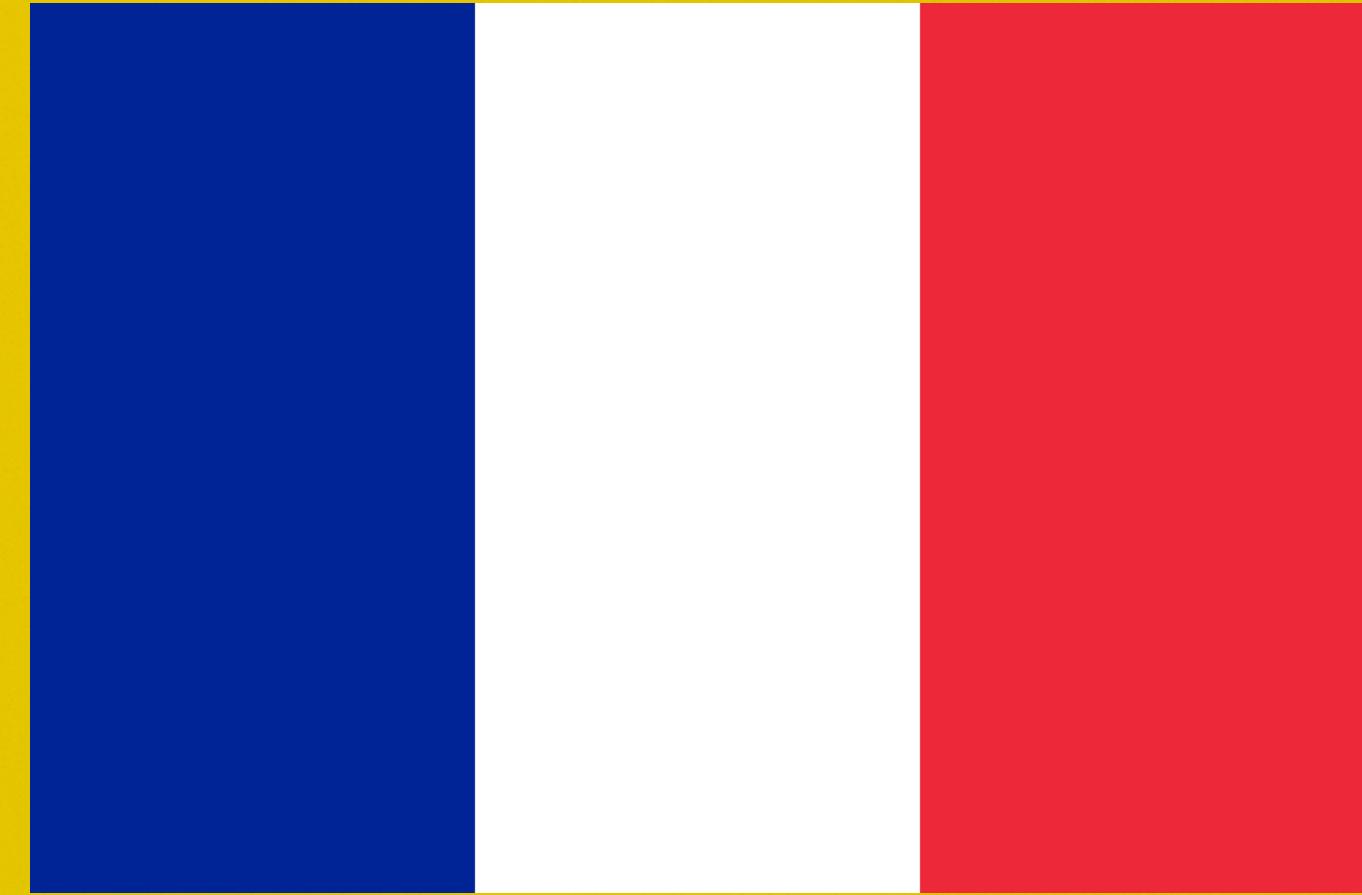

Final Year Individual Project

USING ANSWER SET GRAMMARS FOR TEXT SUMMARIZATION

BACKGROUND



Bilingual
Bi-cultural

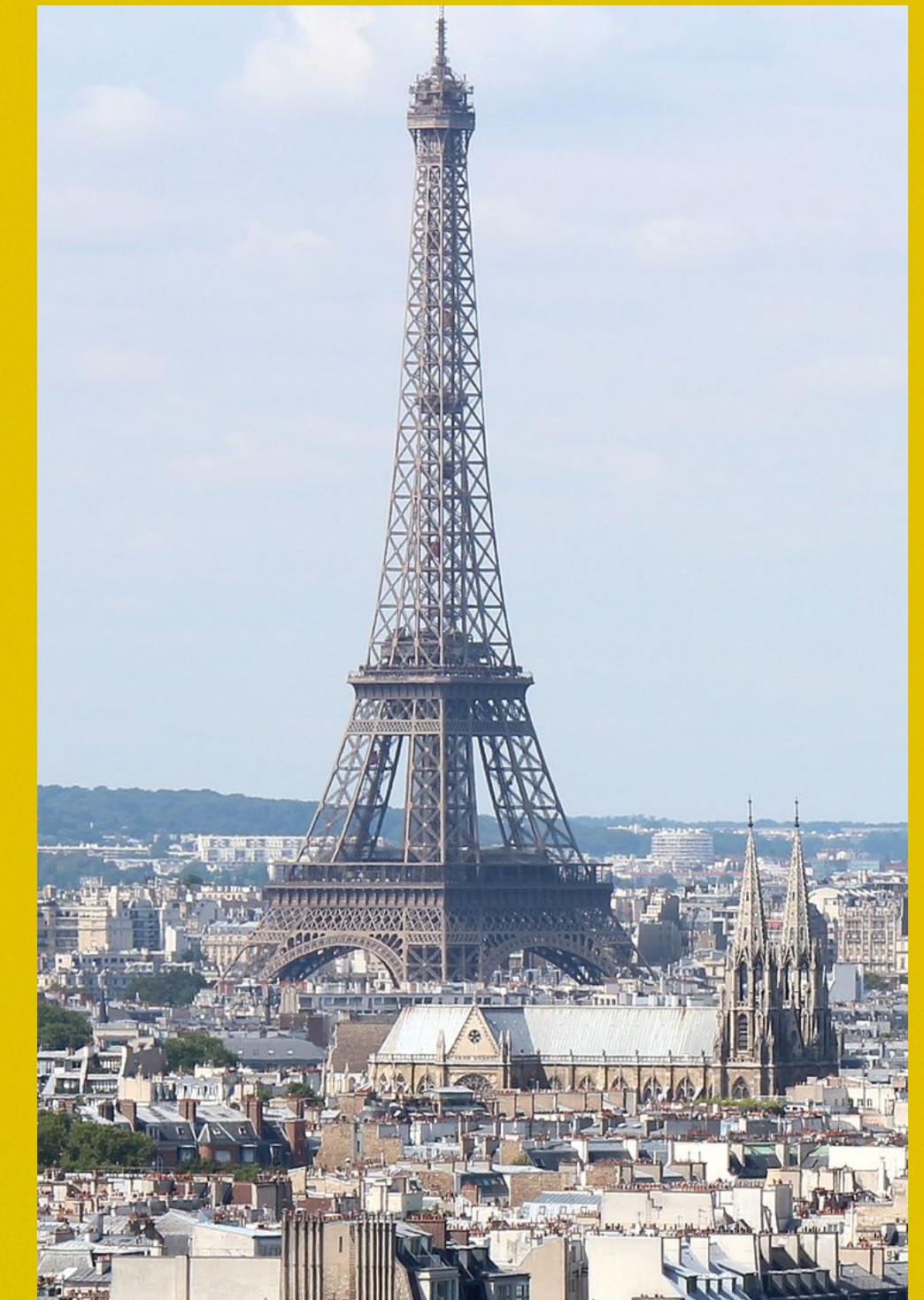
FRENCH AMERICAN





Grew up in

PARIS, FRANCE





Master of Engineering (4 years)
Computer Science

IMPERIAL COLLEGE





TESLA

Summer 2018 (12 weeks)
Summer 2019 (5 months)

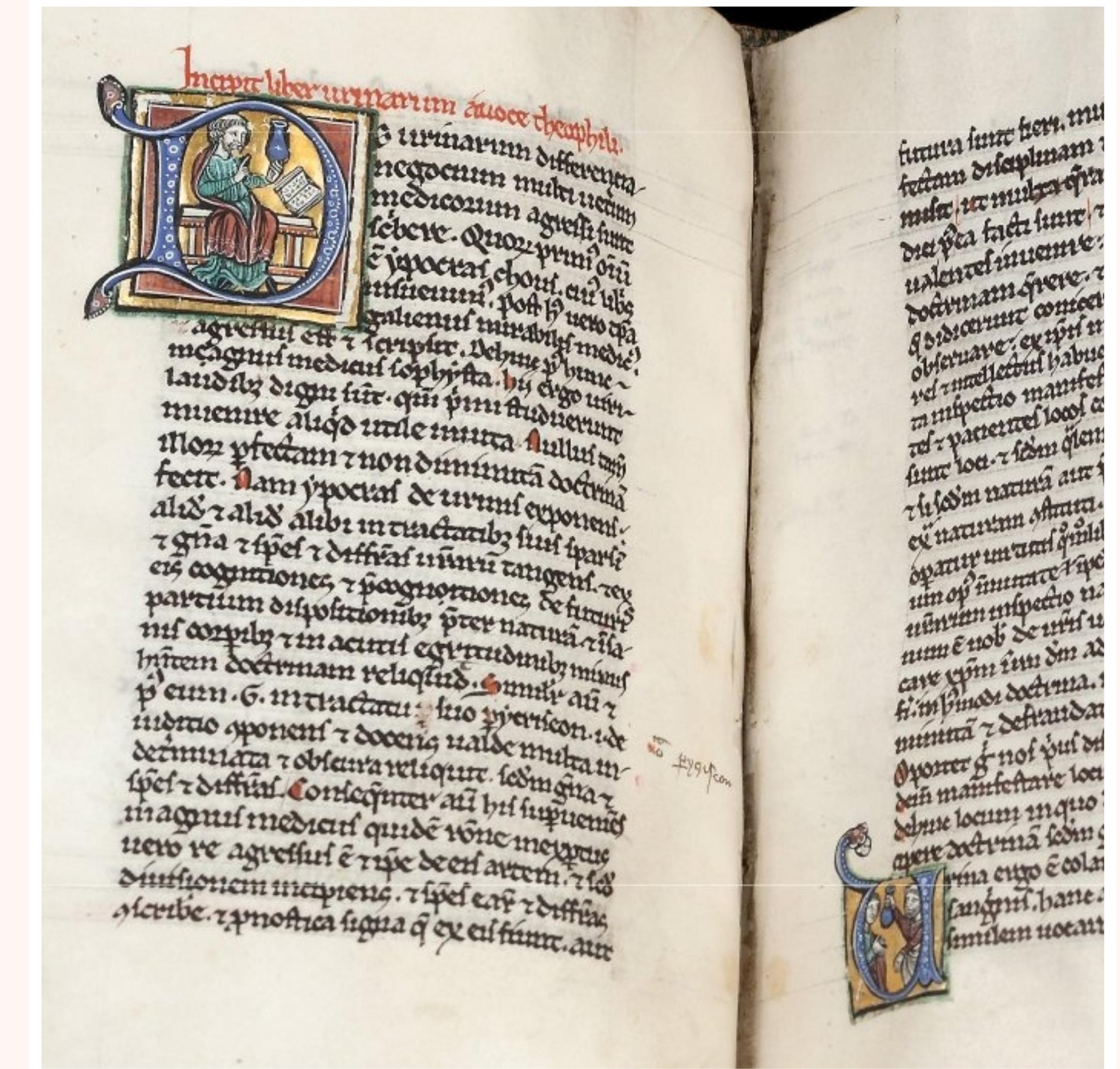
TESLA INTERNSHIP



PROJECT

WHAT IS TEXT SUMMARIZATION?

- Extract most important information from text
- Rewrite in more concise form
- Using at most half as many words



SUMMARIZATION: EXAMPLE

- **There was a curious little boy named Peter Little. He was interested in stars and planets. So he was serious in school and always did his homework. When he was older, he studied mathematics and quantum physics. He studied hard for his exams and became an astrophysicist. Now he is famous.**
- A. **Peter Little was interested in space so he studied hard and became a famous astrophysicist.**
- B. **Peter Little was curious about astronomy. He was always serious in school, and now he is famous.**

AUTOMATIC TEXT SUMMARIZATION

- Many possible applications
- Abstract creation
- Headline generation
- Very difficult computational problem
- Ambiguous definition of summary



PROJECT CHALLENGES

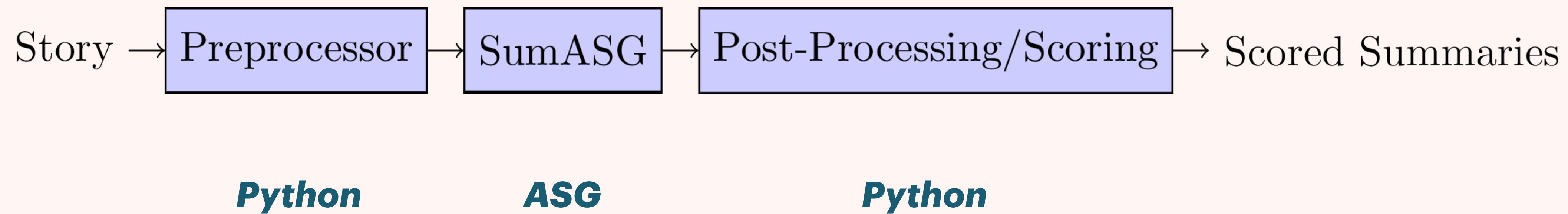
- Open-ended NLP problem
- Usually solved using Machine Learning
- Solution needs to be robust to noise and generate grammatically-correct output



WHAT WAS ACCOMPLISHED?

Logic-based text summarization framework for short stories

Stages of SumASG*



ANSWER SET GRAMMARS¹ (ASG)

- **Context-sensitive grammars where semantic information is expressed as Answer set programs (ASP)**
- **Symbolic learner (ILASP) can be used to learn semantic information missing in incomplete grammar from examples**

¹ Developed by the Spike group at Imperial in 2019: <https://aaai.org/ojs/index.php/AAAI/article/view/4147>

EXAMPLE

start -> nnp vbd jj {}

nnp -> "Peter" {}

nnp -> "Mary" {}

vbd -> "was" {}

jj -> "lazy" {}

jj -> "curious" {}

jj -> "serious" {}

Peter was lazy

Peter was curious

Peter was serious

Mary was lazy

Mary was curious

Mary was serious

EXAMPLE

start -> nnp vbd jj {}

nnp -> "Peter" {} hard_worker.

nnp -> "Mary" {}

vbd -> "was" {}

jj -> "lazy" {}

jj -> "curious" {} hard_worker.

jj -> "serious" {} hard_worker.

ASG: EXAMPLE

start -> nnp vbd jj {}

nnp -> "Peter" { hard_worker. }

nnp -> "Mary" {}

vbd -> "was" {}

jj -> "lazy" {}

jj -> "curious" { hard_worker. }

jj -> "serious" { hard_worker. }

Examples

+ ["Peter", "was", "curious"]
+ ["Peter", "was", "serious"]
- ["Peter", "was", "lazy"]

+ ["Mary", "was", "lazy"]

#modeba(hard_worker).

Mode bias

EXAMPLE

start -> nnp vbd jj { :- hard_worker@1, not hard_worker@3. }

nnp -> "Peter" { hard_worker. }

nnp -> "Mary" {}

vbd -> "was" {}

jj -> "lazy" {}

jj -> "curious" { hard_worker. }

jj -> "serious" { hard_worker. }

ASG: EXAMPLE

Examples

- ? ENGLISH IS VERY COMPLICATED
- ⇒ GRAMMAR COULD BECOME HUGE
- ⇒ SOLUTION: SIMPLIFY INPUT

JJ -> serious { haru_worker.s

Mode bias

PREPROCESSOR

IDEA:

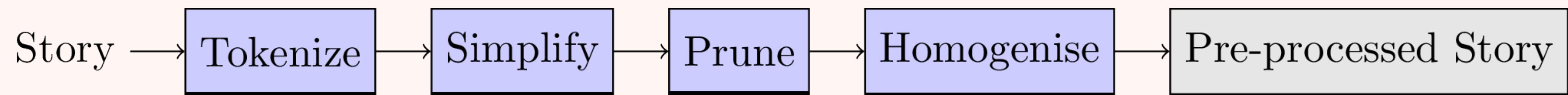
**CONVERT STORY INTO CANONICAL
SUBJECT-VERB-OBJECT STRUCTURE**

+

MINIMISE VOCABULARY DIVERSITY

PREPROCESSOR

Stage 1



There was a curious little boy named Peter Little. He was interested in stars and planets. So he was serious in school and always did his homework. When he was older, he studied mathematics and quantum physics. He studied hard for his exams and became an astrophysicist. Now he is famous.

STEP 1: TOKENIZE

- Separate text into tokens
- Assign part of speech (POS) tag to each token

[('There', 'EX'),
('was', 'VBD'),
('a', 'DT'),
('curious', 'JJ'),
('little', 'JJ'),
('boy', 'NN'),
('named', 'VBN'),
('Peter', 'NNP'),
('Little', 'NNP'),
('.', '.')]

There was a curious little boy named Peter Little. He was interested in stars and planets. So he was serious in school and always did his homework. When he was older, he studied mathematics and quantum physics. He studied hard for his exams and became an astrophysicist. Now he is famous.

STEP 2: SIMPLIFY

- Transform into sentences following canonical subject-verb-object structure

[('There', 'EX'),
('was', 'VBD'),
('a', 'DT'),
('curious', 'JJ'),
('little', 'JJ'),
('boy', 'NN'),
('.', '.')]

('named', 'VBN'),
('Peter', 'NNP'),
('Little', 'NNP'),
('.', '.')]

[('There', 'EX'), S
('was', 'VBD'), V

('a', 'DT'), O
('curious', 'JJ'), O

('little', 'JJ'), O
('boy', 'NN'), O

('.', '.'), O

[('the', 'DT'), S
('curious', 'JJ'), O

('little', 'JJ'), O
('boy', 'NN'), O

('was', 'VBD'), V
('named', 'VBN'), V

('Peter', 'NNP'), O
('Little', 'NNP'), O

('.', '.'), O

There was a curious little boy. the curious little boy was named Peter Little. Peter Little was interested in astronomy. Peter Little was serious in school. PeterLittle did homework always. Peter Little studied mathematics and quantum-physics. Peter Little studied for exams hard. Peter Little became a astrophysicist. Peter Little is famous Now.

STEP 2: SIMPLIFY

- Transform into sentences following canonical subject-verb-object structure
- Combine compound nouns into single tokens

[('There', 'EX'),
('was', 'VBD'),
('a', 'DT'),
('curious', 'JJ'),
('little', 'JJ'),
('boy', 'NN'),
('.', '.')]

('named', 'VBN'),
('Peter', 'NNP'),
('Little', 'NNP'),
('.', '.')]

[('There', 'EX'), S
('was', 'VBD'), V

('a', 'DT'),
('curious', 'JJ'),
('little', 'JJ'),
('boy', 'NN'),
('.', '.')]

[('the', 'DT'),
('curious', 'JJ'),
('little', 'JJ'),
('boy', 'NN'), S

('was', 'VBD'), V

('named', 'VBN'),
('PeterLittle', 'NNP'), O
('.', '.')]

There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was interested in astronomy. PeterLittle was serious in school. PeterLittle did homework always. PeterLittle studied mathematics and quantum-physics. PeterLittle studied for exams hard. PeterLittle became a astrophysicist. PeterLittle is famous Now.

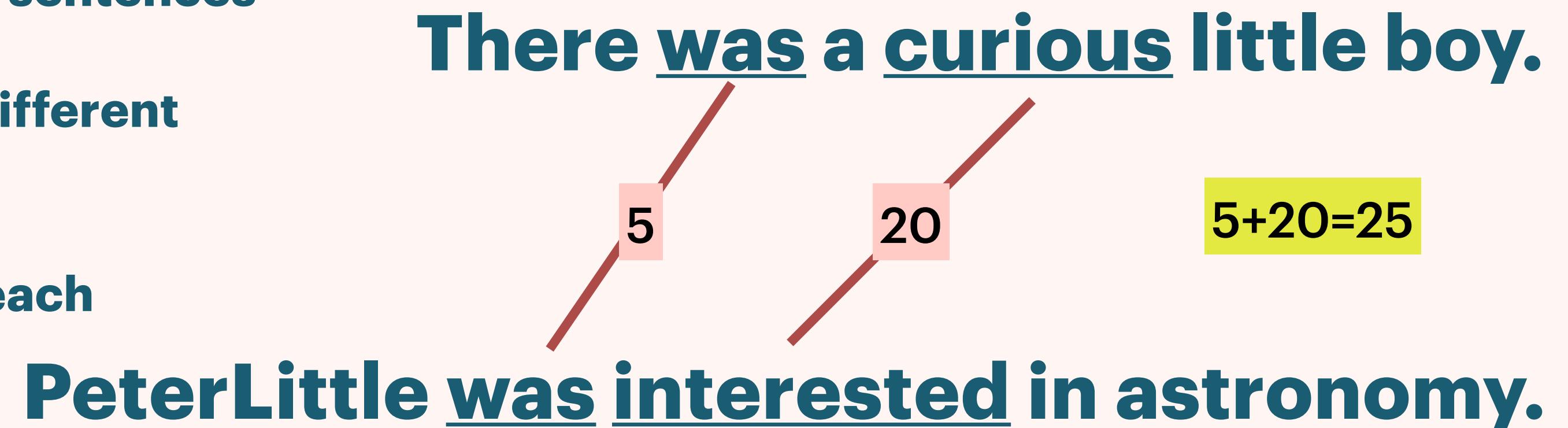
STEP 3: PRUNE

- **Goal: remove irrelevant sentences to speed up SumASG and help it produce better summaries**

There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was interested in astronomy. PeterLittle was serious in school. PeterLittle did homework always. PeterLittle studied mathematics and quantum-physics. PeterLittle studied for exams hard. PeterLittle became a astrophysicist. PeterLittle is famous Now.

STEP 3: PRUNE

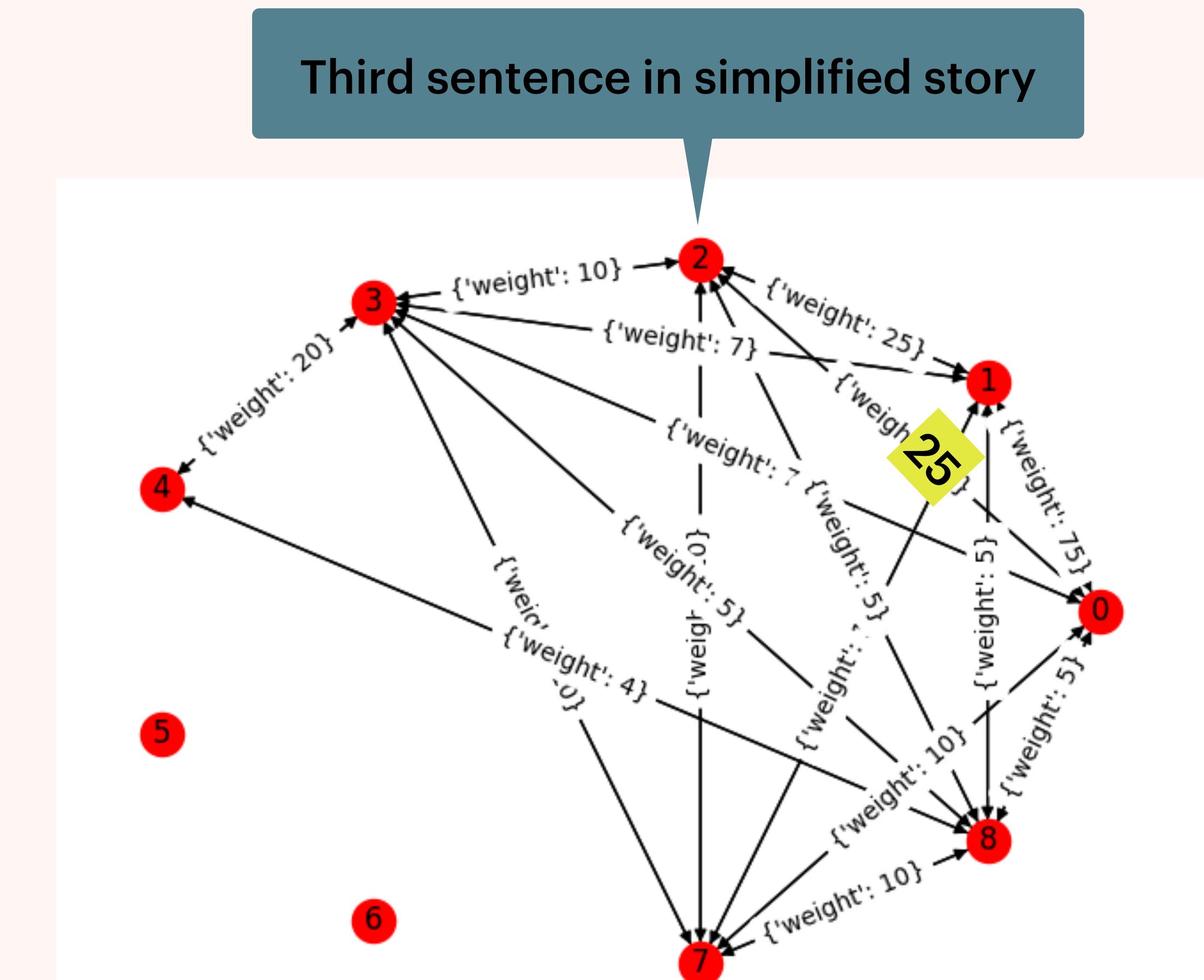
- **Goal:** remove irrelevant sentences to speed up SumASG and help it produce better summaries
- Generate text relationship map (TRM) for sentences
 - Look up similarity between words of different sentences (using API)
 - Compute sum of similarity points for each sentence pair



There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was interested in astronomy. PeterLittle was serious in school. PeterLittle did homework always. PeterLittle studied mathematics and quantum-physics. PeterLittle studied for exams hard. PeterLittle became a astrophysicist. PeterLittle is famous Now.

STEP 3: PRUNE

- **Goal:** remove irrelevant sentences to speed up SumASG and help it produce better summaries
 - Generate text relationship map (TRM) for sentences
 - Look up similarity between words of different sentences (using API)
 - Compute sum of similarity points for each sentence pair

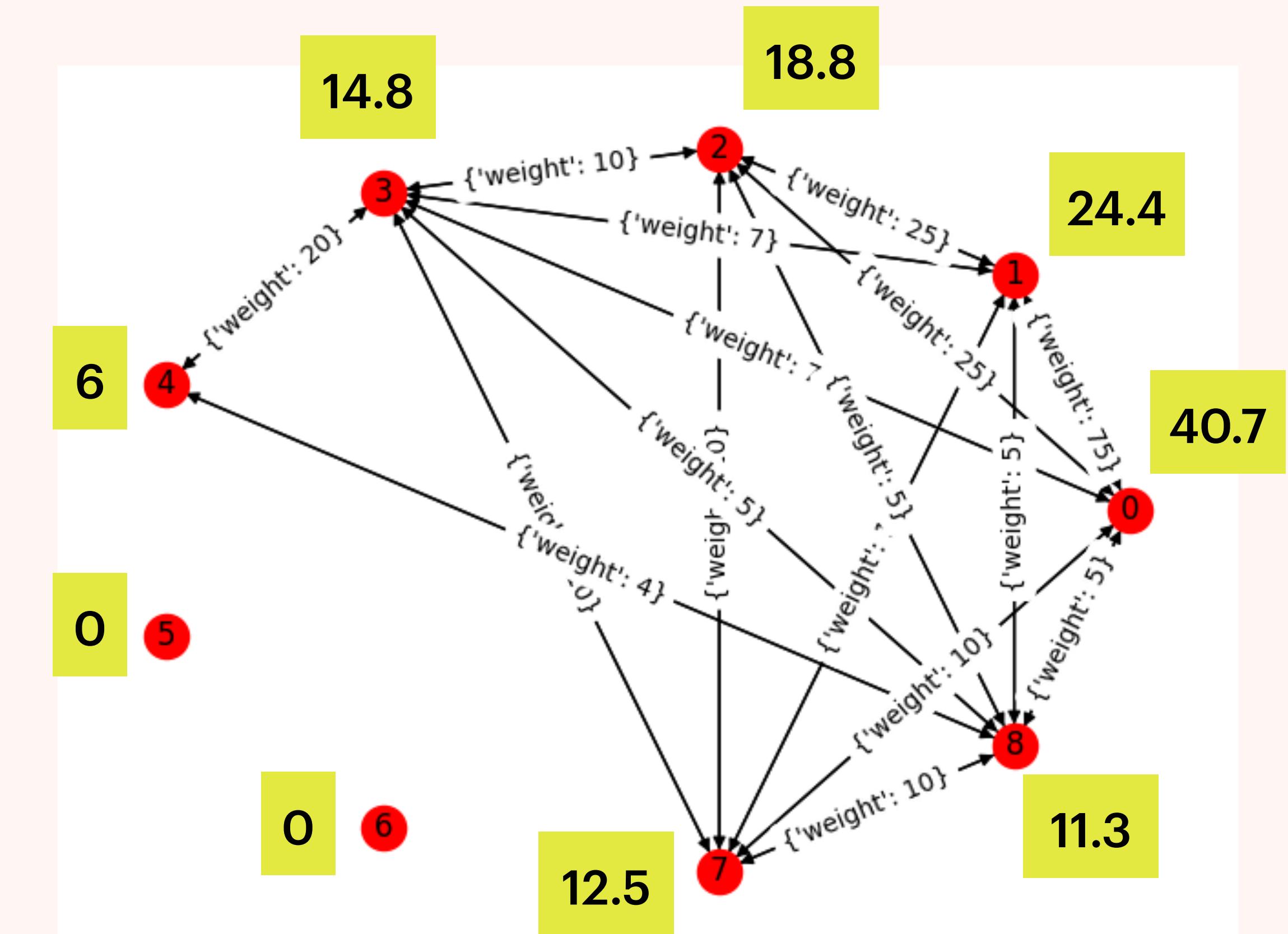


0 There was a curious little boy.
ittle was interested in astronomy.

There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was interested in astronomy. PeterLittle was serious in school. PeterLittle did homework always. PeterLittle studied mathematics and quantum-physics. PeterLittle studied for exams hard. PeterLittle became a astrophysicist. PeterLittle is famous Now.

STEP 3: PRUNE

- **Goal: remove irrelevant sentences to speed up SumASG and help it produce better summaries**
- **Compute importance for each sentence**
- **Sum weight of outgoing arcs in TRM**
- **Divide result to normalise according to sentence length**



There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was interested in astronomy. PeterLittle was serious in school. PeterLittle did homework always. PeterLittle studied mathematics and quantum-physics. PeterLittle studied for exams hard. PeterLittle became a astrophysicist. PeterLittle is famous Now.

STEP 3: PRUNE

- **Goal: remove irrelevant sentences to speed up SumASG and help it produce better summaries**
- **Order sentences by importance**
- **Prune sentences with importance below 25th percentile**

Q1 = 6.0

('There was a curious little boy.', 40.7)
(~~'the curious little boy was named PeterLittle.'~~, 24.4)
(~~'PeterLittle was interested in astronomy.'~~, 18.8)
(~~'PeterLittle was serious in school.'~~, 14.8)
(~~'PeterLittle became a astrophysicist.'~~, 12.5)
(~~'PeterLittle is famous Now.'~~, 11.3)
(~~'PeterLittle did homework always.'~~, 6.0)
(~~'PeterLittle studied mathematics and quantum-physics.'~~, 0.0)
(~~'PeterLittle studied for exams hard.'~~, 0.0)

There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was interested in astronomy. PeterLittle was serious in school. PeterLittle did homework always. PeterLittle became a astrophysicist. PeterLittle is famous Now.

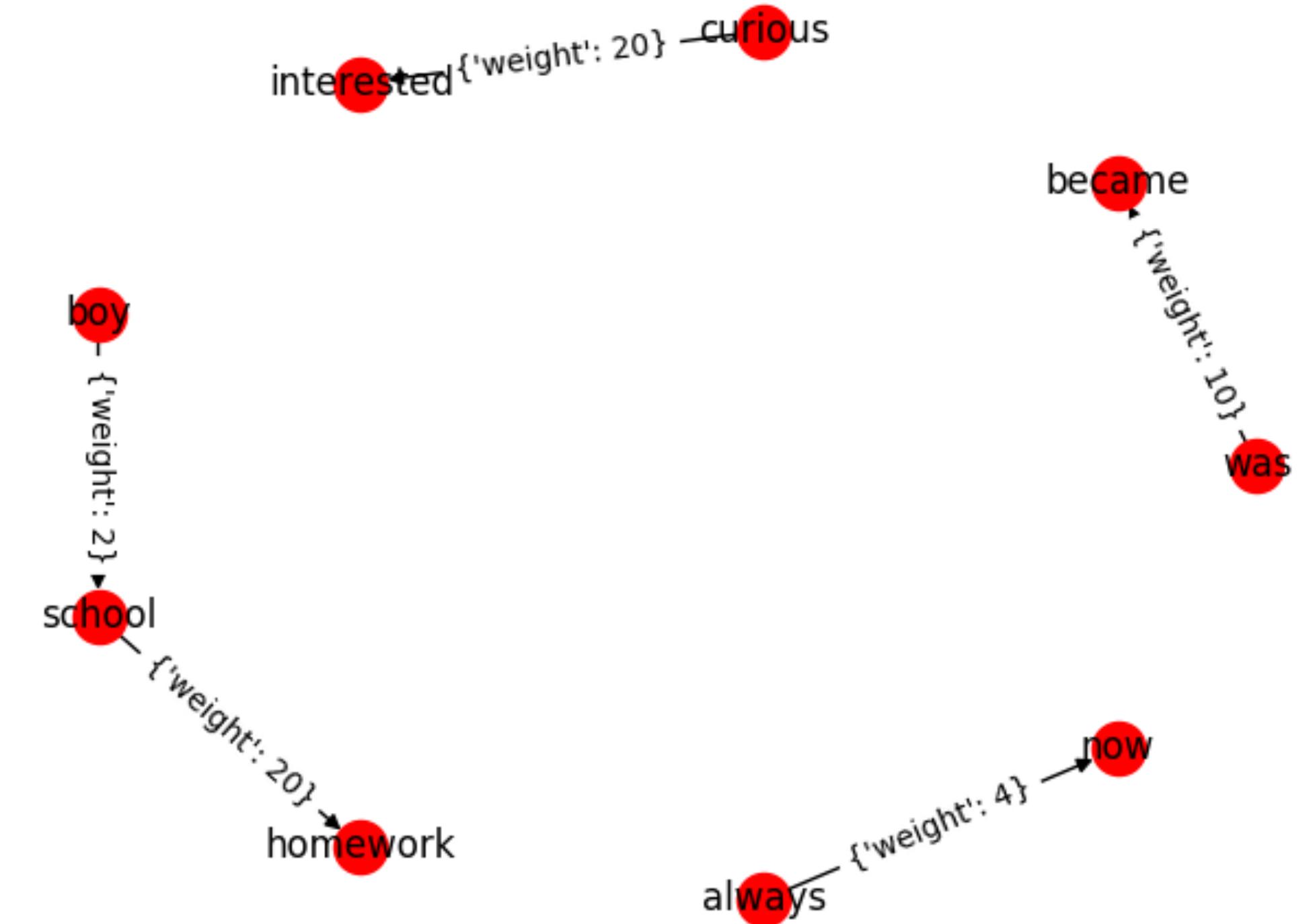
STEP 4: HOMOGENISE

- **Goal: reduce lexical diversity by avoiding use of synonyms**

There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was interested in astronomy. PeterLittle was serious in school. PeterLittle did homework always. PeterLittle became a astrophysicist. PeterLittle is famous Now.

STEP 4: HOMOGENISE

- **Goal:** reduce lexical diversity by avoiding use of synonyms
- Find word pairs with high similarity (≥ 20)



[{'curious', 'interested'}, {'homework', 'school'}]

There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was interested in astronomy. PeterLittle was serious in school. PeterLittle did homework always. PeterLittle became a astrophysicist. PeterLittle is famous Now.

STEP 4: HOMOGENISE

- **Goal:** reduce lexical diversity by avoiding use of synonyms
- Replace all synonyms with unique representative for each set of synonyms

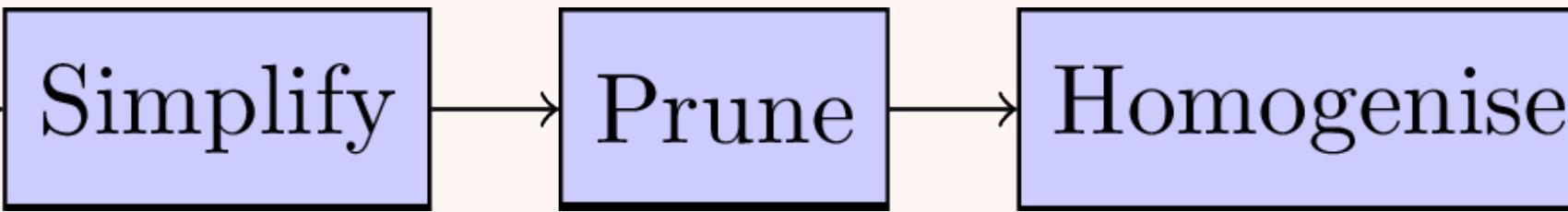
There was a curious little boy.
the curious little boy was named PeterLittle.
PeterLittle was interested curious in astronomy.
PeterLittle was serious in school.
PeterLittle did homework school always.
PeterLittle became a astrophysicist.
PeterLittle is famous Now.

[{'curious', 'interested'}, {'homework', 'school'}]

PREPROCESSOR

Stage 1

There was a curious little boy named Peter Little. He was interested in stars and planets. So he was serious in school and always did his homework. When he was older, he studied mathematics and quantum physics. He studied hard for his exams and became an astrophysicist. Now he is famous.



There was a curious little boy. the curious little boy was named PeterLittle. PeterLittle was curious in astronomy. PeterLittle was serious in school. PeterLittle did school always. PeterLittle became a astrophysicist. PeterLittle is famous Now.

Word count	Unique words
51	36

Word count	Unique words
35	20

SUMASG

INTERNAL REPRESENTATION: LEAF NODES

- Represent words in ASG using production rules
 - LHS: POS tag
 - RHS: string
- Encode sentence position using predicates and constants in logic

nnp -> "PeterLittle" { noun(peterlittle). }

vbd -> "was" { verb(be,past). }

jj -> "curious" { adj_or_adv(curious). }

in -> "in" { det(in). }

nn -> "astronomy" { noun(astronomy). }

INTERNAL REPRESENTATION: NON-LEAF NODES

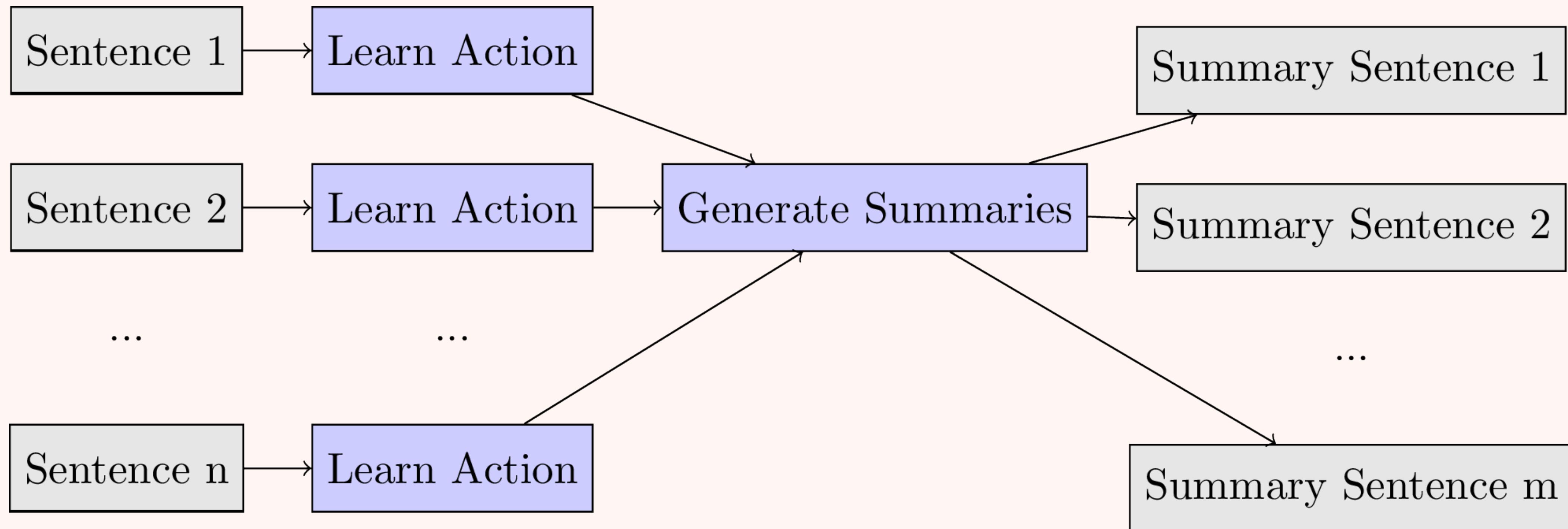
- Represent groups of words in ASG using production rules
- LHS: noun/verb part (np/vp) or sentence (s)
- RHS: POS tags or np/vp
- Push leaf node predicates up from the bottom of the parse tree

```
np -> jj in nn {  
    object(N,D,A) :- adj_or_adv(A)@1,  
    det(D)@2,  
    noun(N)@3.  
}
```

```
vp -> vbd np {  
    verb(N,T) :- verb(N,T)@1.  
    object(N,D,A) :- object(N,D,A)@2.  
}
```

SUMASG

Stage 2



IN

+ ["PeterLittle ", "was ", "curious ",
"in ", "astronomy ", ". "]

STEP 1: LEARN ACTIONS

- **Goal:** translate story into internal representation using abduction through ILASP
- Define mode bias to learn facts from input sentences
- With a learning task semantic information from strings comes for free

```
#modeh(action(  
    verb(const(main_verb),const(main_form)),  
    subject(const(noun),const(det),const(adj_or_adv)),  
    object(const(noun),const(det),const(adj_or_adv)))).
```

```
#constant(adj_or_adv,curious).  
#constant(det,in).  
#constant(main_form,past).  
#constant(main_verb,be).  
#constant(noun,astronomy).  
#constant(noun,peterlittle).
```

STEP 1: LEARN ACTIONS

- **Goal:** translate story into internal representation using abduction through ILASP
- Add special constraint in production rule for sentences that enforces learnable predicate **action(2, verb(be, past), "action" to correspond to words in subtree**

IN

+ ["PeterLittle ", "was ", "curious ",
"in ", "astronomy ", ". "]

s -> np vp {

: - not action(verb(V_N,V_T),
subject(S_N,S_D,S_A),
object(O_N,O_D,O_A)),
verb(V_N,V_T)@2,
subject(S_N,S_D,S_A)@1,
object(O_N,O_D,O_A)@2.

}

OUT

**action(2, verb(be, past),
subject(peterlittle, 0, 0),
object(astronomy, in , curious))**

STEP 1: LEARN ACTIONS

- **Goal:** translate story into internal representation using abduction through ILASP
- Repeat learning task for each sentence in story



...

**action(2, verb(be, past),
subject(peterlittle, 0, 0),
object(astronomy, in, curious)).**

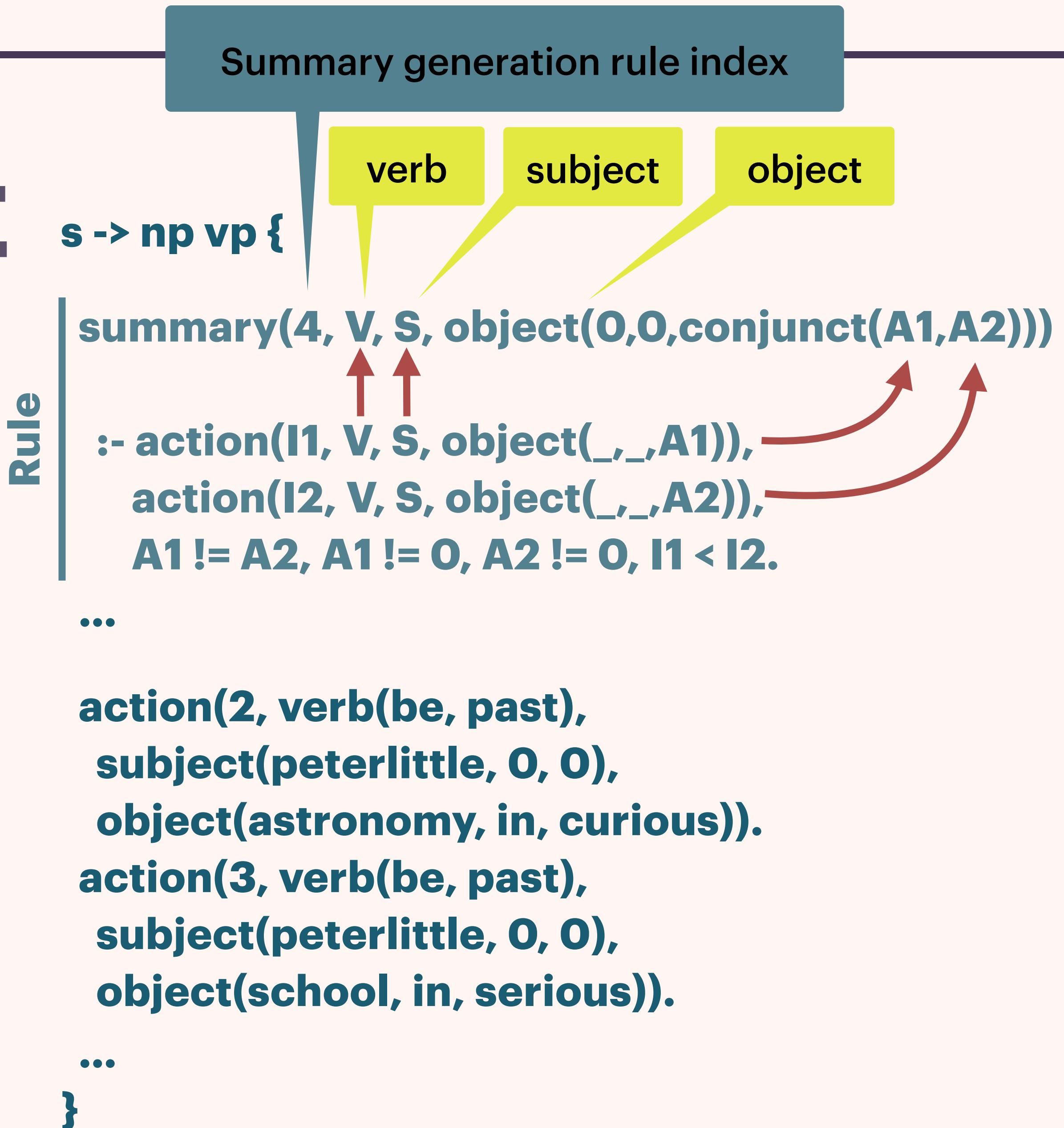
**action(3, verb(be, past),
subject(peterlittle, 0, 0),
object(school, in, serious)).**

**action(4, verb(do, past),
subject(peterlittle, 0, 0),
object(school, 0, always)).**

...

STEP 2: GENERATE SUMMARY SENTENCES

- **Goal: compress learned actions into sentences for summary**
- **Add learned actions to production rule for sentences**
- **Create set of summary generation rules to form summary sentences from actions**



STEP 2: GENERATE SUMMARY SENTENCES

- **Goal:** compress learned actions into sentences for summary
- Run augmented grammar containing summary generation rules to obtain summary sentences

**action(2, verb(be, past),
subject(peterlittle, 0, 0),
object(astronomy, in, curious)).**

**action(3, verb(be, past),
subject(peterlittle, 0, 0),
object(school, in, serious)).**

**summary(4,verb(be, past),
subject(peterlittle, 0, 0),
object(0, 0,
conjunct(curious, serious))).**

OUT

PeterLittle was curious and serious .

STEP 2: GENERATE SUMMARY SENTENCES

OUT

- **Goal: compress learned actions into sentences for summary**
- **Run augmented grammar containing summary generation rules to obtain summary sentences**
- **Thanks to ASG translation from internal representation into strings comes for free**

there was a curious little boy .

the curious little boy was named PeterLittle .

PeterLittle was curious in astronomy .

PeterLittle was serious in school .

PeterLittle was curious and serious .

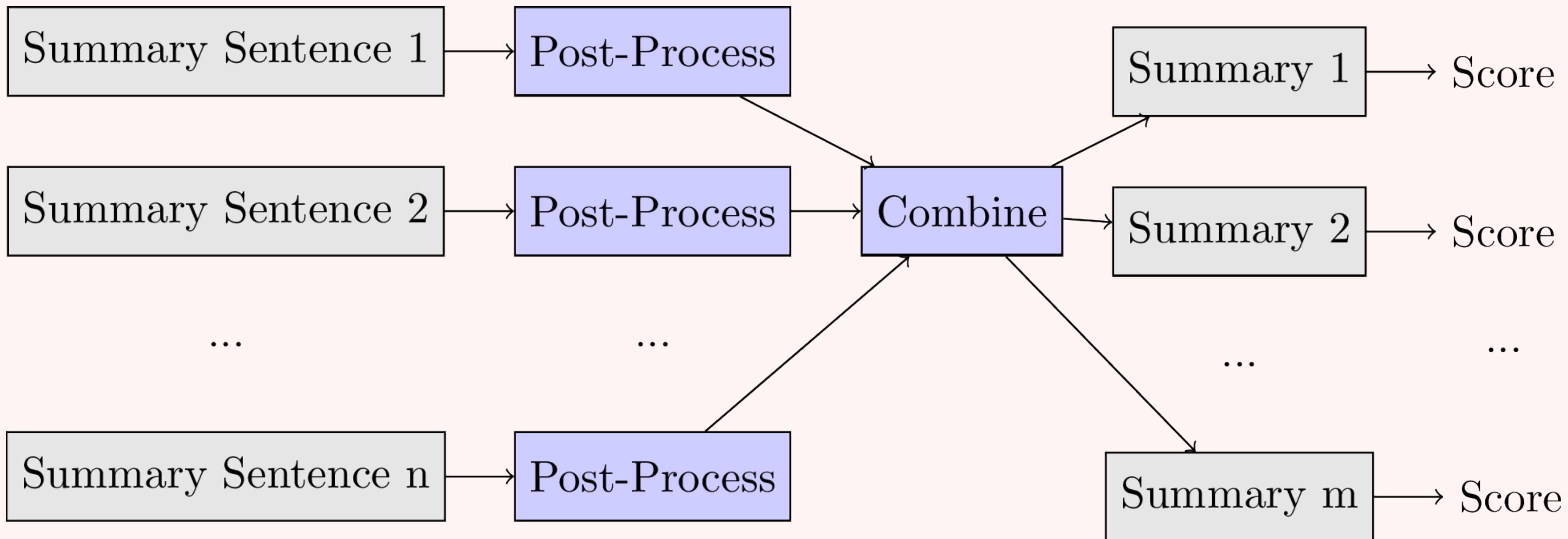
PeterLittle did school always .

PeterLittle is famous now .

POST-PROCESSING / SCORING

POST-PROCESSING / SCORING

Stage 3



STEP 1: POST- PROCESS AND COMBINE

- **Revert Preprocessor simplifications to make summary sentences grammatically correct**

```
there was a curious little boy .  
the curious little boy was named PeterLittle .  
PeterLittle was curious in astronomy .  
PeterLittle was serious in school .  
PeterLittle was curious and serious .  
PeterLittle did school always .  
PeterLittle is famous now .
```

There was a curious little boy.

The curious little boy was named Peter Little.

Peter Little was curious in astronomy.

Peter Little was serious in school.

Peter Little was curious and serious.

Peter Little did school always.

Peter Little is famous now.

STEP 1: POST-PROCESS AND COMBINE

- Generate all order-preserving summary sentence combinations of length 3 (or less) to create summaries

For the list [0, 1, 2, 3], these would be:

- [0, 1, 2]
- [0, 2, 3]
- [1, 2, 3]

There was a curious little boy.
The curious little boy was named Peter Little.
Peter Little was curious in astronomy.
Peter Little was serious in school.
Peter Little was curious and serious.
Peter Little did school always.
Peter Little is famous now.

There was a curious little boy. Peter Little was curious in astronomy. Peter Little did school always.

Peter Little was curious in astronomy. Peter Little was serious in school. Peter Little was curious and serious.

Peter Little was serious in school. Peter Little did school always. Peter Little is famous now.

...

There was a curious little boy named Peter Little. He was interested in stars and planets. So he was serious in school and always did his homework. When he was older, he studied mathematics and quantum physics. He studied hard for his exams and became an astrophysicist. Now he is famous.

STEP 2: SCORING

- **Goal:** quantify quality of generated summaries
- Compute enhanced version of type-token ratio (TTR) for each summary to measure information density
 - Count unique words in summary
 - Divide result by total number of words
 - For all counts ignore frequent words (e.g., "a", "is") and main topics (e.g., "Peter Little")

('Peter Little is famous now. There was a curious little boy. Peter Little did school always.', 17)

('Peter Little is famous now. Peter Little did school always. Peter Little was curious in astronomy.', 17)

('Peter Little is famous now. There was a curious little boy. Peter Little was serious in school.', 17)

('Peter Little did school always. Peter Little was curious in astronomy. Peter Little was serious in school.', 13)

...

There was a curious little boy named Peter Little. He was interested in stars and planets. So he was serious in school and always did his homework. When he was older, he studied mathematics and quantum physics. He studied hard for his exams and became an astrophysicist. Now he is famous.

STEP 2: SCORING

- **Goal: quantify quality of generated summaries**
- **Keep only summaries with score above 75th percentile (top summaries)**

('Peter Little is famous now. There was a curious little boy. Peter Little did school always.', 17)

('Peter Little is famous now. Peter Little did school always. Peter Little was curious in astronomy.', 17)

('Peter Little is famous now. There was a curious little boy. Peter Little was serious in school.', 17)

Q3 = 15.0

~~('Peter Little did school always. Peter Little was curious in astronomy. Peter Little was serious in school.', 13)~~

...

There was a curious little boy named Peter Little. He was interested in stars and planets. So he was serious in school and always did his homework. When he was older, he studied mathematics and quantum physics. He studied hard for his exams and became an astrophysicist. Now he is famous.

STEP 2: SCORING

- **Goal:** quantify quality of generated summaries
- If reference summary is given assert BLEU score with one of the top summaries is high

('Peter Little is famous now. Peter Little did school always. Peter Little was curious in astronomy.', 17)

References

- A. **Peter Little was curious about astronomy.**
He was always serious in school, and now he is famous. BLEU = 0.66
- B. **Peter Little was curious about astronomy.**
He was always serious in school, and now he is famous. BLEU = 0.4

EVALUATION

IDEA:

**USE GENERATIVE MODEL TO
CREATE TRAINING CORPUS**

+

TRAIN ENCODER-DECODER

TRAINING CORPUS: EXAMPLE

STORY

- A. **The nervous system processed a fraud. The nervous system processed a whale. The nervous system processed the American.**
- B. **The heavy traffic transported the birthday. It was the isolated birthday.**

SUMMARY

- A. **The nervous system processed a fraud and a whale. The nervous system processed the American.**
- B. **The heavy traffic transported the isolated birthday.**

VALIDATION

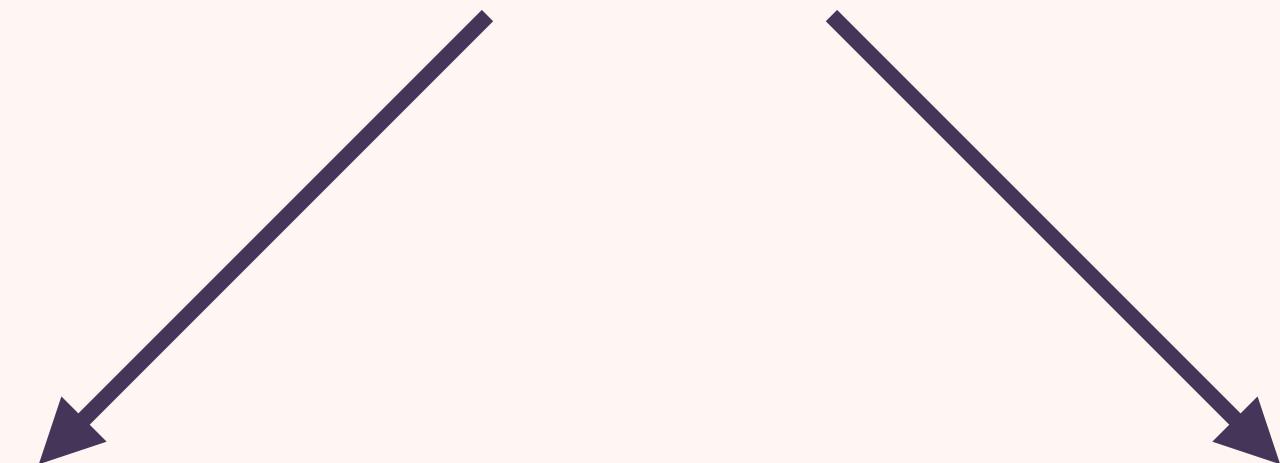
- Encoder-decoder was able to learn summary generation rules used by SumASG
- Discrepancy in final validation accuracy due to multiplicity of top summaries (e.g., different ordering possible for conjunctions)

Epochs	20	100	400
Training accuracy (%)	60.68	98.73	99.83
Validation accuracy (%)	61.54	90.24	92.07
Training perplexity	12.45	1.06	1.01
Validation perplexity	16.73	2.07	2.36
Test perplexity	-	-	2.05

EVALUATION: EXPERIMENT 1

- **Goal:** compare robustness to perturbations
- Move adjective from subject to object
 - **SumASG***: produces same adjective-free summary
 - **Neural network**: uses tokens not present in input

The system processed a nervous fraud. The system processed a nervous whale. The system processed the nervous American.



The system processed
a **banking** and a **killer**.
The system processed
the **nervous**.

Encoder-decoder

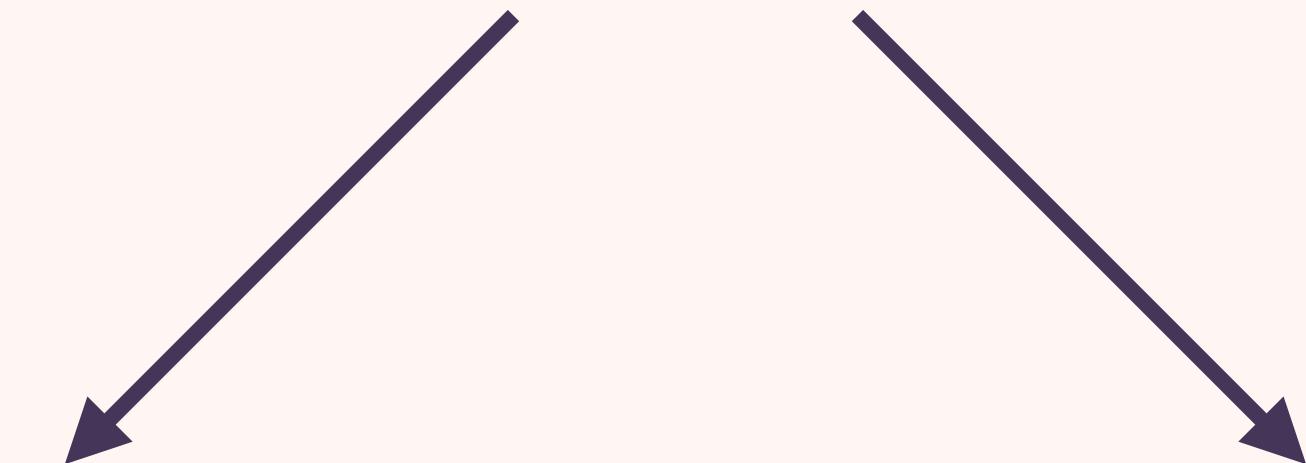
The system processed
a **fraud** and a **whale**.
The system processed
the **nervous American**.

*SumASG**

EVALUATION: EXPERIMENT 1

- **Goal:** compare robustness to perturbations
- Swap subject and object in first sentence
 - **SumASG***: generates summary with correct subject-object order
 - Neural network: uses tokens not present in input and changes meaning of story

A fraud processed the nervous system. The nervous system processed a whale. The nervous system processed the American.



**The system processed a banking and a killer.
The system processed the nervous.**

Encoder-decoder

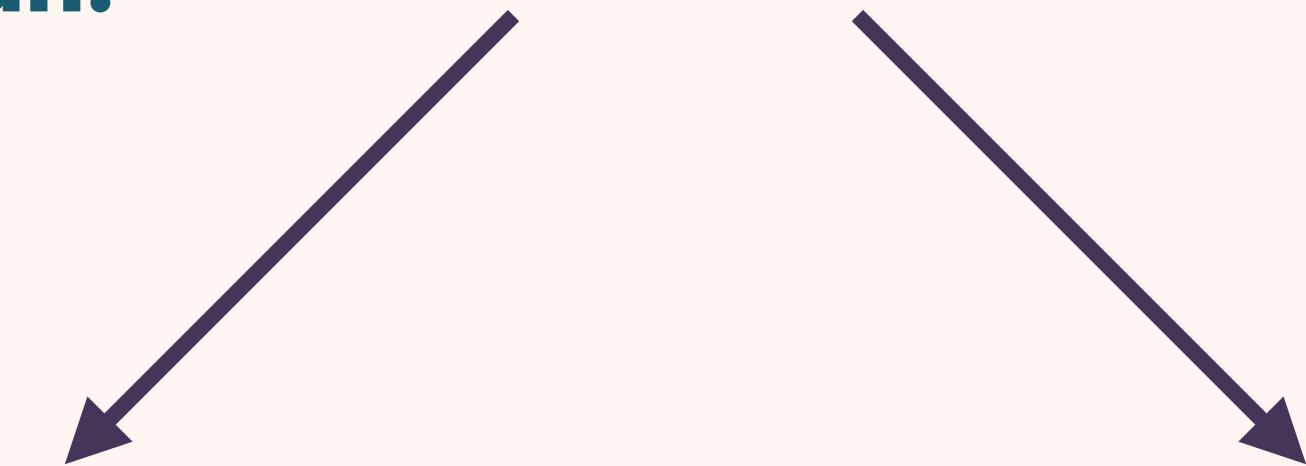
The nervous system processed a whale. A fraud processed the nervous system.

SumASG*

EVALUATION: EXPERIMENT 2

- **Goal:** compare ability to check input validity
- Change word order to make grammatically incorrect
 - SumASG*: rejects input
 - Neural network: ignores invalidity of story and uses tokens not present in input

The system processed processed a nervous fraud. processed The system a nervous whale. a The system processed nervous American.



The system processed
a nervous and a killer.
The system processed
a pregnancy.

<invalid>

Encoder-decoder

SumASG*

EVALUATION: EXPERIMENT 2

- **Goal:** compare ability to check input validity
- Create random word sequence
- SumASG*: rejects input
- Neural network: ignores invalidity of story and produces meaningless sequence

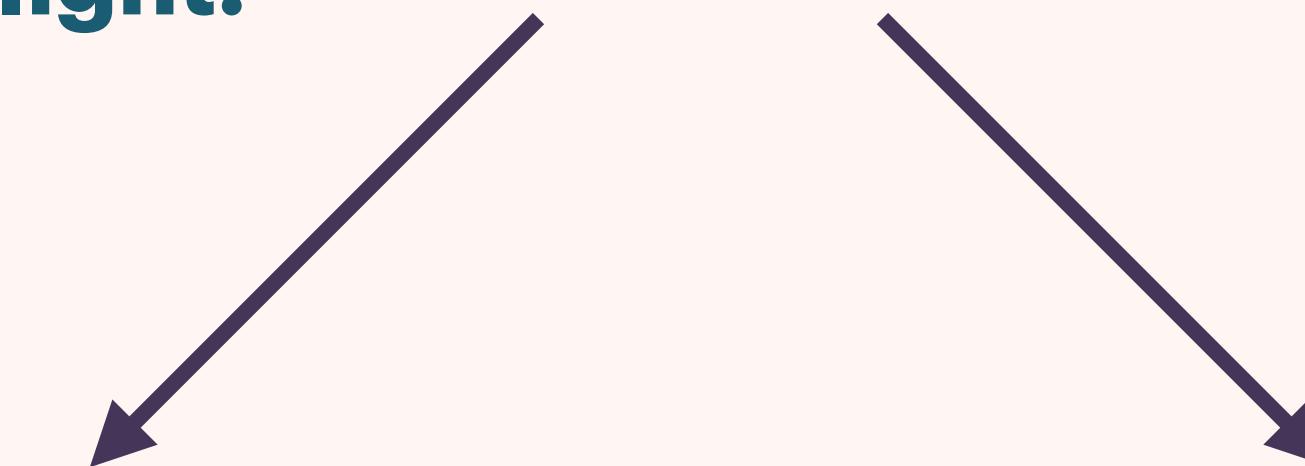
question hunger whole ruled cleared. needle
front pound spun rented programming.
bought journalism disclosed broad check
died delight.

The

<invalid>

Encoder-decoder

SumASG*



OTHER APPROACHES TO TEXT SUMMARIZATION

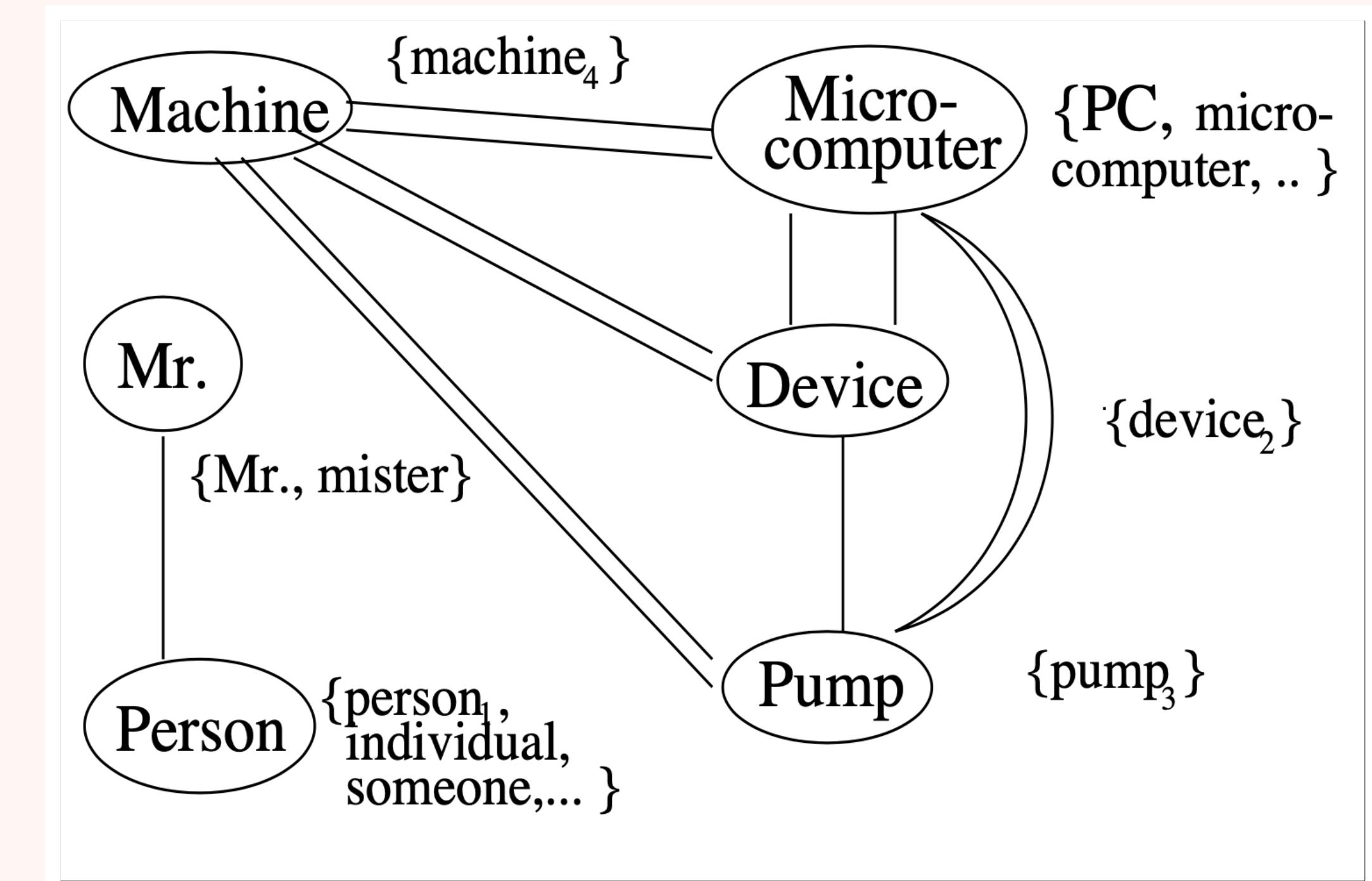
COMBINATORY CATEGORICAL GRAMMAR (CCG)

- Efficient parsing mechanism
- Uses lambda calculus
- Different sequences can evaluate to same representation

$$\frac{\begin{array}{c} \text{Harry} & \text{sees} & \text{Sally} \\ \hline NP_{3s} & (S \setminus NP_{3s}) / NP & NP \\ : harry & : \lambda x \lambda y. sees xy & : sally \end{array}}{\frac{\longrightarrow}{\begin{array}{c} S \setminus NP_{3s} \\ : \lambda y. sees sally y \end{array}}} \frac{\longrightarrow}{\begin{array}{c} S \\ : sees sally harry \end{array}}$$

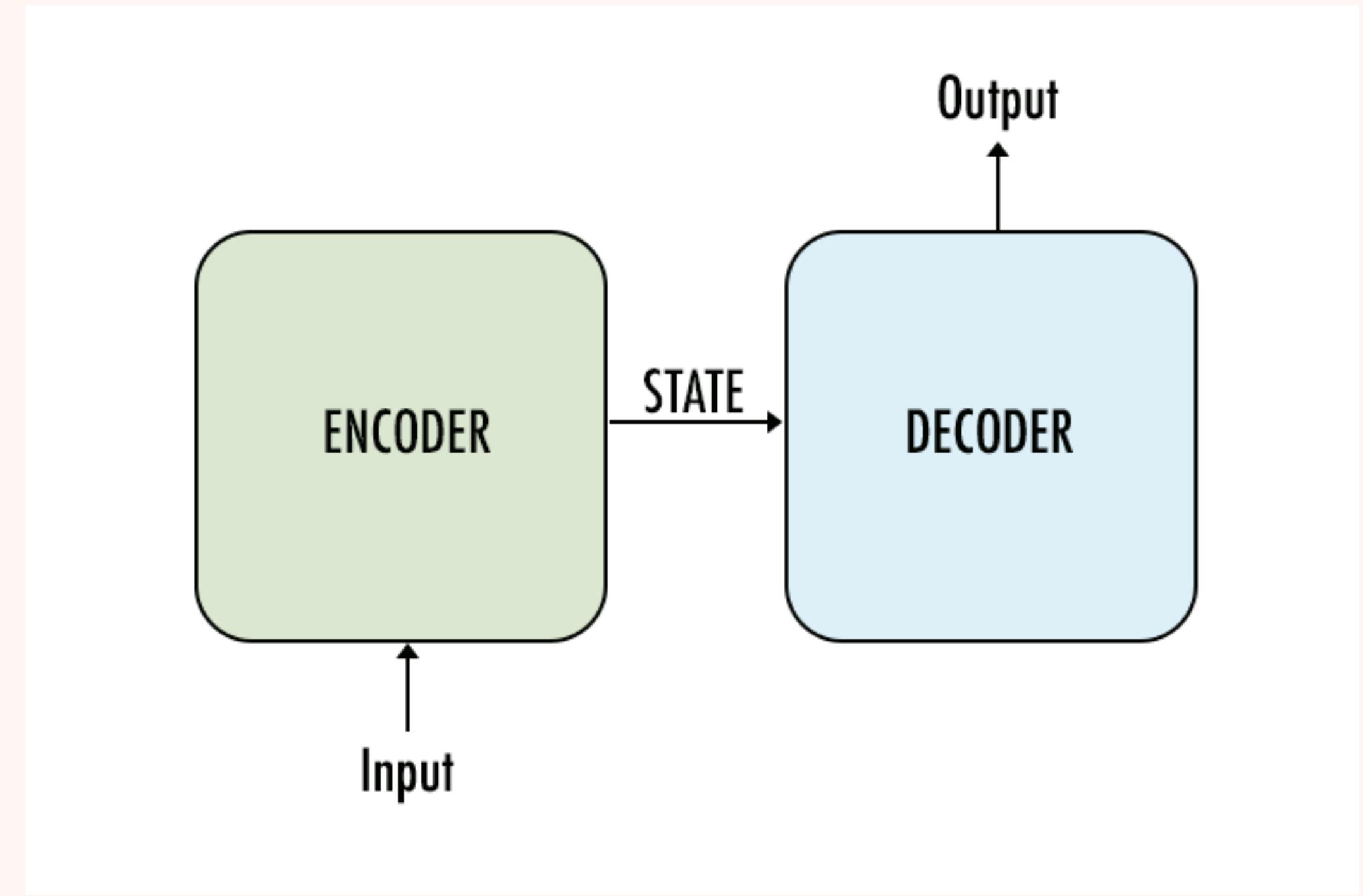
LEXICAL CHAINS

- Create chains of semantically-related words in text
- Choose "strongest" lexical chain (based on word count and distribution across text)
- Sentence in story with the most words from the preferred lexical chain becomes summary



ENCODER- DECODERS

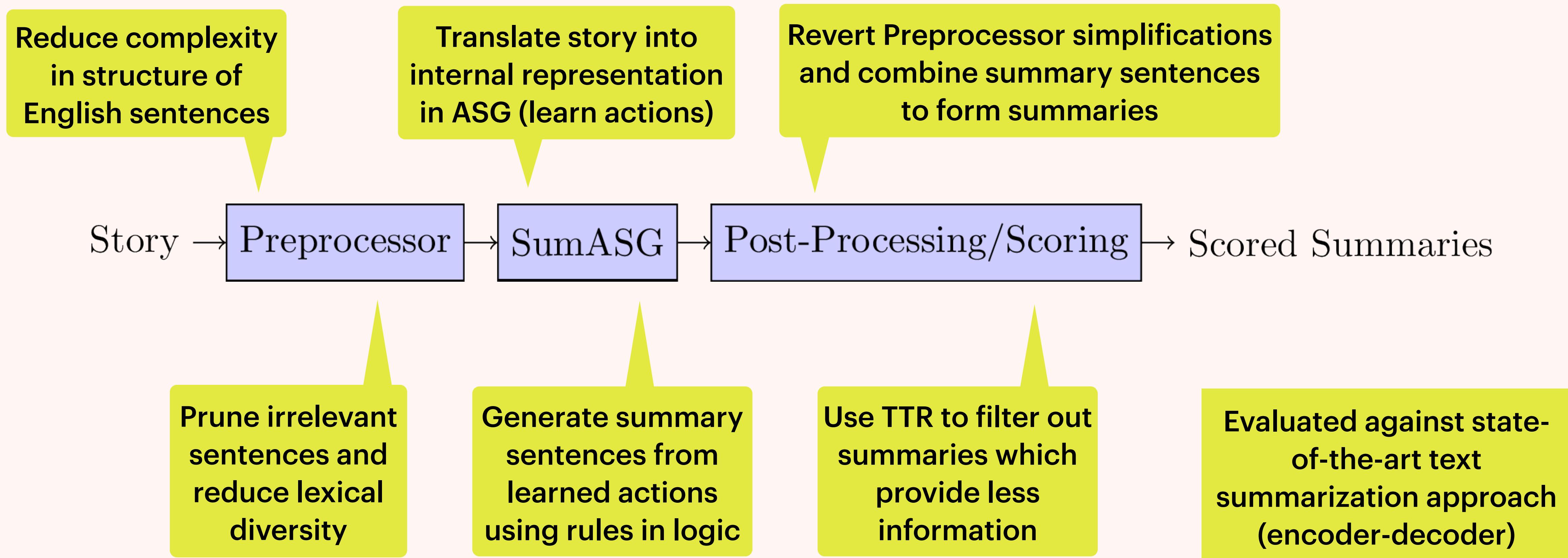
- Capable of learning any rule-based sequence-to-sequence translation
- Highly customisable architecture
- Need to be trained on very large datasets to perform well



CONCLUSION

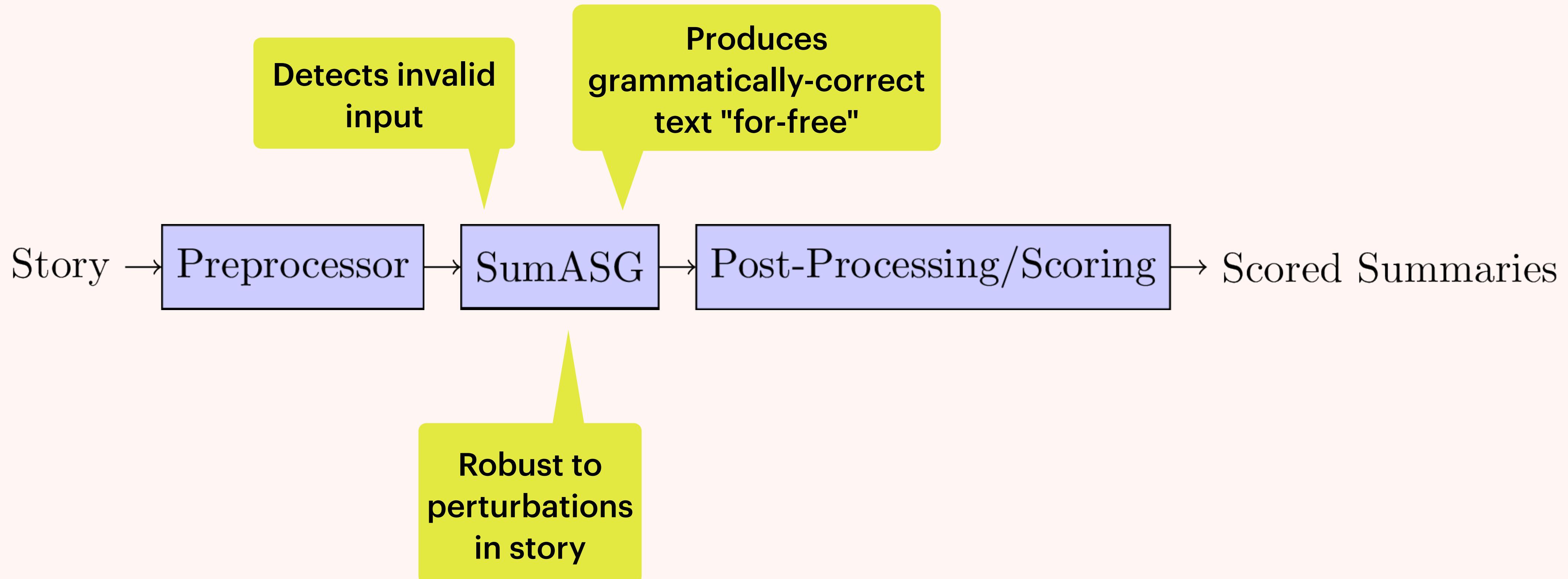
ACCOMPLISHMENTS

Logic-based text summarization framework for short stories

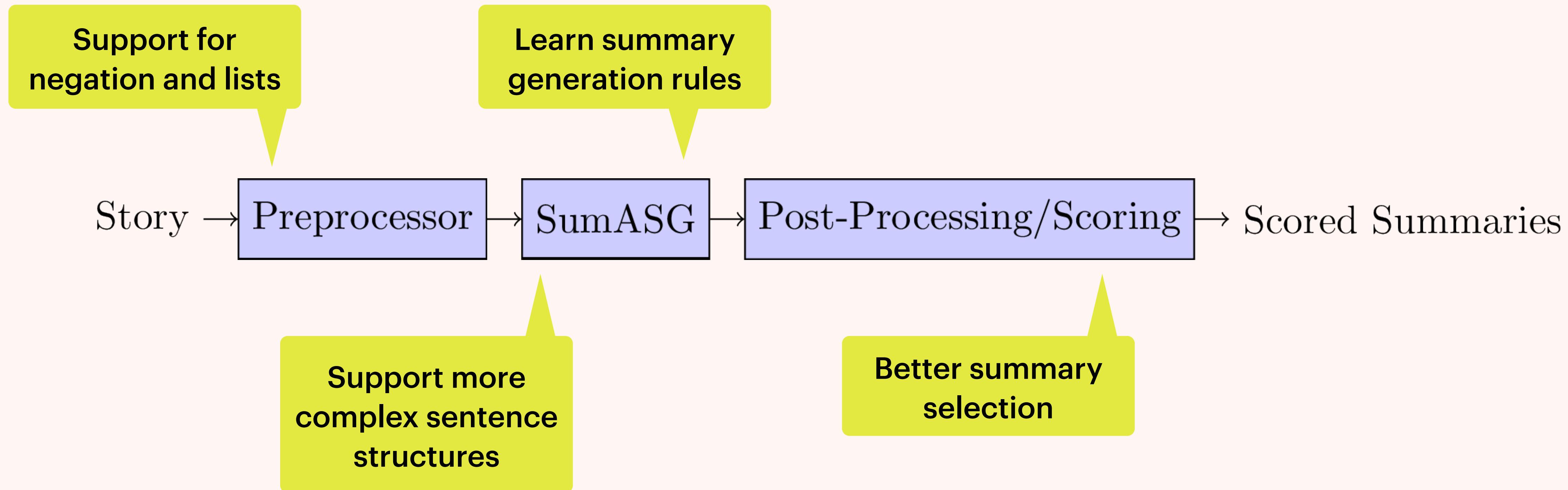


CHALLENGES MET

As seen from evaluation experiments



FUTURE WORK



SOURCES

- https://upload.wikimedia.org/wikipedia/en/thumb/c/c3/Flag_of_France.svg/2560px-Flag_of_France.svg.png
- https://upload.wikimedia.org/wikipedia/en/thumb/a/a4/Flag_of_the_United_States.svg/2560px-Flag_of_the_United_States.svg.png
- https://upload.wikimedia.org/wikipedia/commons/thumb/0/08/Seine_and_Eiffel_Tower_from_Tour_Saint_Jacques_2013-08.JPG/2560px-Seine_and_Eiffel_Tower_from_Tour_Saint_Jacques_2013-08.JPG
- https://upload.wikimedia.org/wikipedia/commons/thumb/6/60/Seine_Pont_Royal_Louvre_Paris.jpg/2560px-Seine_Pont_Royal_Louvre_Paris.jpg
- https://upload.wikimedia.org/wikipedia/commons/5/5d/London_Underground_with_Big_Ben.JPG
- https://media-exp1.licdn.com/dms/image/C4D1BAQGi1T31ocKgKg/company-background_10000/0?e=2159024400&v=beta&t=h5ylw9SZR66kOn_BHodXoe3NpgpogA6L-ABXONaIDGo
- https://en.wikipedia.org/wiki/File:De_urinarum_differencia_negocium_between_1210_and_1230_.JPG
- <https://i.pinimg.com/originals/0f/cc/05/0fcc0590885e92a46a273744f44d7248.jpg>
- https://upload.wikimedia.org/wikipedia/commons/thumb/2/2a/Tesla_Visit_3_%289267529364%29.jpg/1200px-Tesla_Visit_3_%289267529364%29.jpg
- <http://www.sci-news.com/othersciences/linguistics/voynich-manuscript-07190.html>
- <http://homepages.inf.ed.ac.uk/steedman/papers/ccg/moravcsik.pdf>
- <https://doi.org/10.7916/D85B09VZ>
- https://miro.medium.com/max/686/1*3lj8AGqfwEE5KCTJ-dXTvg.png

THANK YOU, QUESTIONS?