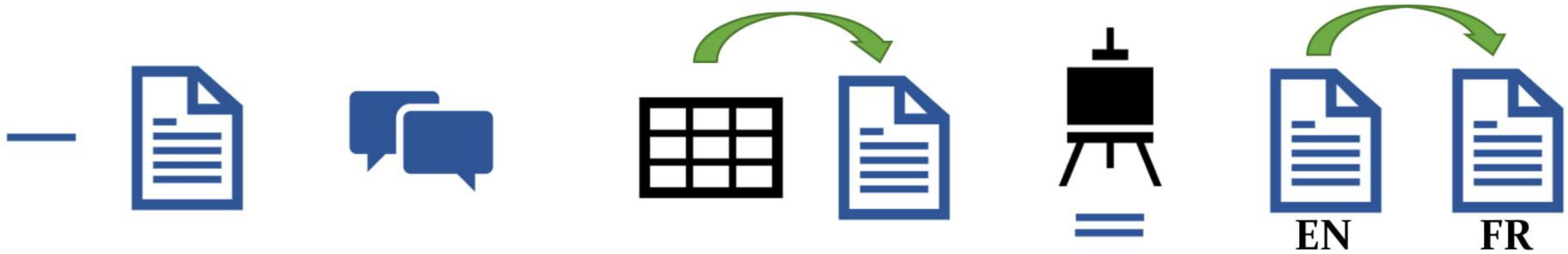
# Natural Language Generation

**Gerasimos Lampouras**
**University of Sheffield**

# Natural language generation

...is the natural language processing task of generating text...

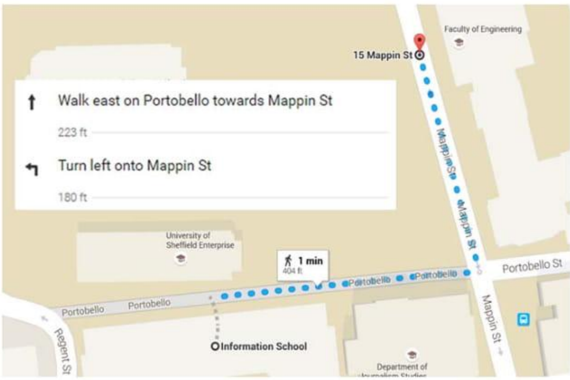



...in order to meet specific communicative goals.

# Applications all around us!

**NLG** can be used

- to present/summarize information,
- to produce repetitive documents,
- in dialog systems / chatbots.




# Problem setup

Training data is natural language sentences.

$$D_{train} = \{x^1, \ldots, x^M\}$$

$$\mathrm{x} = [x_1, \ldots, x_N]$$

***V** is the vocabulary*

*We want to learn a model that can generate sentences!*

# So far...

## Language modelling!

- *How likely is that a sequence of words comes from a particular language / domain (e.g. English news articles)?*

$$P(x) = P(x_1, \dots, x_N)$$

$$= P(x_1)P\left(x_2, \dots, x_N \middle| x_1\right)$$

$$= P(x_1)P\left(x_2 \middle| x_1\right) \dots P\left(x_N \middle| x_1, \dots, x_{N-1}\right)$$

$$= \prod_{n=1}^{N} P(x_n | x_1, \dots, x_{n-1}) \qquad \text{(chain rule)}$$

*k-th order Markov assumption*

$$P(x) = \prod_{n=1}^{N} P(x_n | x_{n-k}, \dots, x_{n-1})$$

$$P(x_n | x_{n-k}, \dots, x_{n-1}) = \frac{counts(x_{n-k}, \dots, x_{n-1}, x)}{counts(x_{n-k}, \dots, x_{n-1})}$$

# LM-based NLG

*How to generate a sentence using a language model?*

- *Given a particular sequence of words:*
  *which is the most likely word to follow?*

| None | The | water | is | clear | . | \n |
| --- | --- | --- | --- | --- | --- | --- |

$$\hat{y} = \underset{x_n \in V}{argmax} P(x_n | x_{n-k}, \ldots, x_{n-1})$$

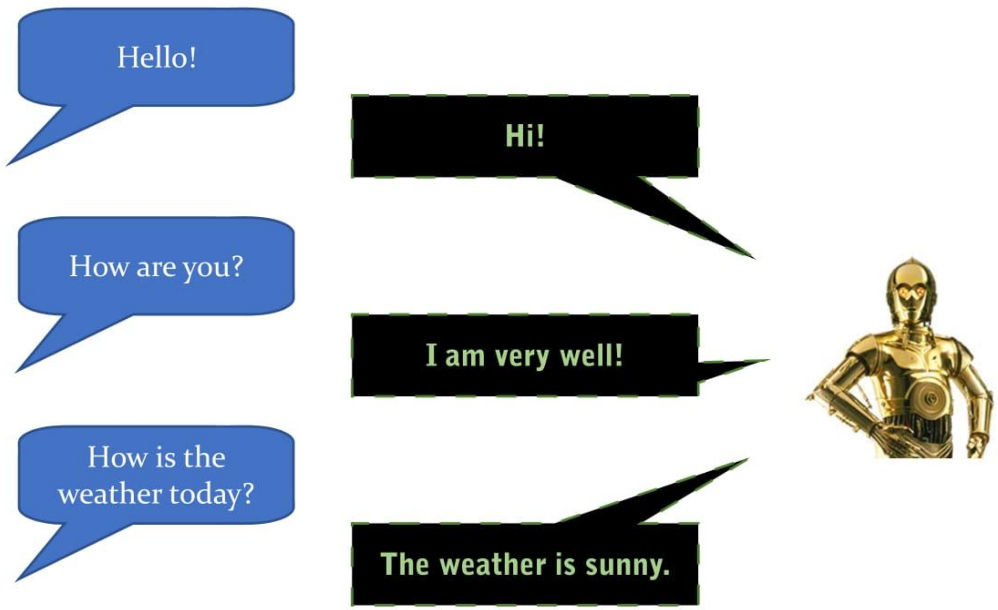
*"This is how you do it: you sit down at the keyboard and you put one word after another until its done. It's that easy, and that hard."*

-Neil Gaiman

# Simple dialogue responses

Language modelling is ungrounded

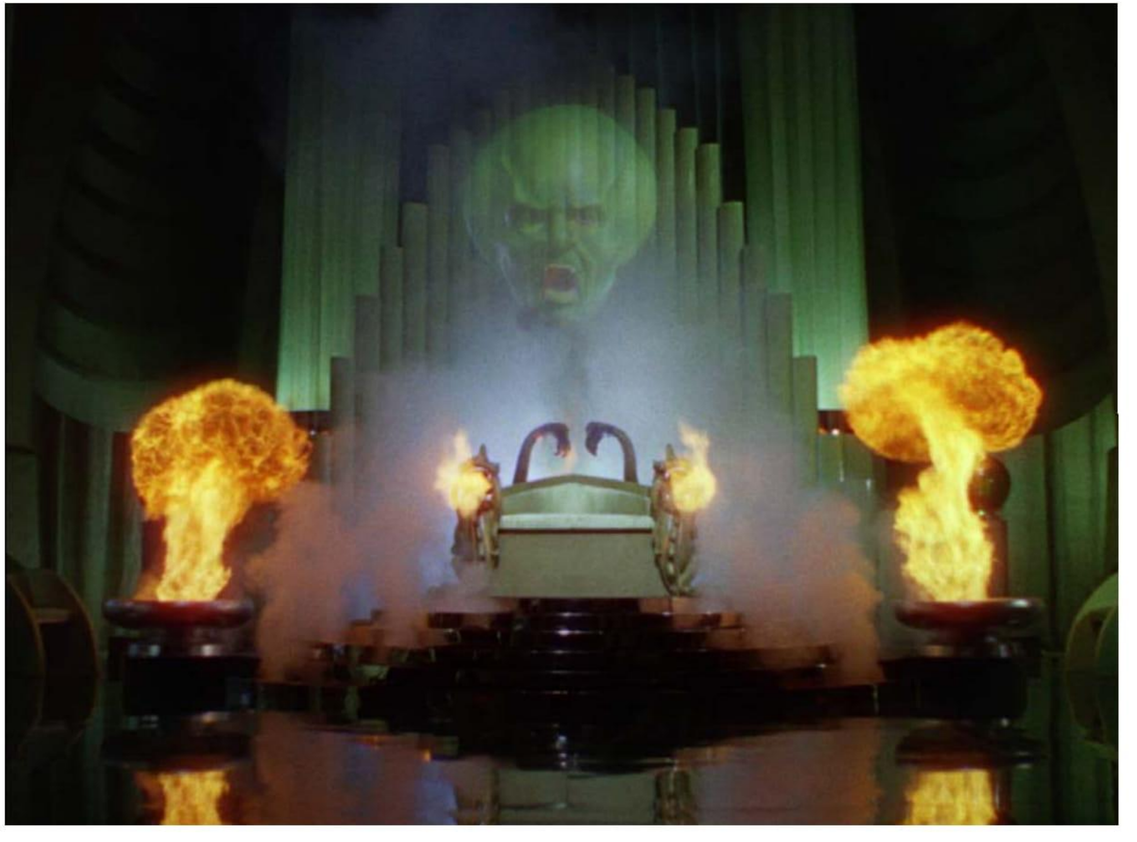
- *What is the communicative goal here?*
- *In open-domain dialogue, the goal is to produce appropriate responses.*



# Problem setup - dialogues

Data won't be simple sentences, but user dialogue input, and responses.

- Can use existing dialogues from messaging services
  (<u>with the participants' consent!</u>)
- Or gather the data through wizard-of-oz experiments.



# Problem setup - dialogues

Data won't be simple sentences, but user dialogue input, and responses.

- Can use existing dialogues from messaging services (with the participants' consent!)
- Or gather the data through wizard-of-oz experiments.

$$D_{train} = \{u^1 : x^1, \dots, u^M : x^M\}$$

$$\text{u} = [u_1, \dots, u_N]$$
$$\text{x} = [x_1, \dots, x_N]$$

***V** is the vocabulary of responses*

*We want to learn a model that can generate responses, given a dialogue history!*

# How to respond?

Condition language models to dialogue history.

$$P(x) = \prod_{n=1}^{N} P(x_n | x_{n-1}, \dots, x_{n-k}; u)$$

How is the weather today?

None | The | weather | is | sunny | ! | \n

# How to respond?

Condition language models to dialogue history.

$$P(x) = \prod_{n=1}^{N} P(x_n | x_{n-1}, \dots, x_{n-k}; u)$$

How is the weather today?

None | The | weather | is | sunny | ! | \n

Unlikely to encounter every possible dialogue history ***u*** during training

$$P(weather|None, The; \text{"}what's\ the\ weather\ like\ ?\text{"}) = \frac{counts(None, The, weather; \text{"}what's\ the\ weather\ like\ ?\text{"})}{counts(None, The; \text{"}what's\ the\ weather\ like\ ?\text{"})}$$

- Represent the dialogue history as feature vectors.

# How to respond?

Condition language models to dialogue history.

$$P(x) = \prod_{n=1}^{N} P(x_n | x_{n-1}, \dots, x_{n-k}; u)$$

How is the weather today?

None | The | weather | is | sunny | ! | \n

Unlikely to encounter every possible dialogue history ***u*** during training

$$P(weather|None, The; \text{"}what's\ the\ weather\ like\ ?\text{"}) = \frac{counts(None, The, weather; \text{"}what's\ the\ weather\ like\ ?\text{"})}{counts(None, The; \text{"}what's\ the\ weather\ like\ ?\text{"})}$$

- Represent the dialogue history as feature vectors.
  - e.g. frequent words: weather, today
  - e.g. communicative goal: ?
- Alternatively use more complicated models: HMM, perceptron, neural


The generated text may be an appropriate response to the user, but is still not considering any ground knowledge.
How is the weather today?
The weather is sunny.
...

# Concept-to-text

Generating text from a machine-readable form of information that we can communicate to the user:

- a meaning representation,
- a set of database records,
- a graph,
- etc.

temperature(min=48,mean=53,max=61)
windSpeed(min=3,mean=6,max=11)
rainChance(mode=highChance)

*"You don't write because you want to say something.*
*You write because you have something to say."*

-F. Scott Fitzgerald

# Meaning representations

```
:StEmilion, isA, :Bordeaux
:StEmilion, :hasColor, :red
:StEmilion, :hasFlavor, :strong
```

→ *St. Emilion is a red strong Bordeaux.*

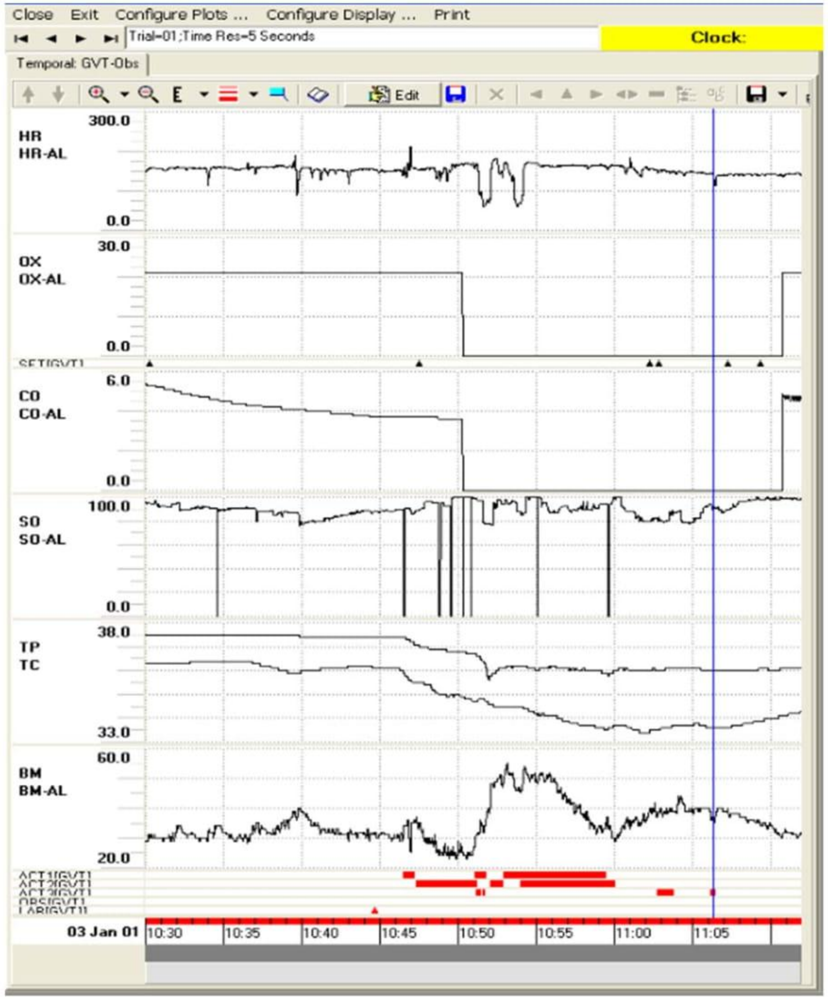
```
?select(
    price_range='cheap or expensive')
```

→ *Sorry, would you like something in the cheap price range or in the expensive price range?*

```
greeting()
```

→ *Hi!*

# Meaning representations




Neonatal Intensive Care Unit data as time series

*By 11:00 the baby had been hand-bagged a number of times causing 2 successive bradycardias. She was successfully re-intubated after 2 attempts. The baby was sucked out twice. At 11:02 FIO2 was raised to 79%.*

# Problem setup – concepts

Data consists of pairs of meaning representations and NL references.

| | | | |
|---|---|---|---|
| name="Cotto"; | food=Fast food; | near="Café Rouge"; | *Cotto is a Fast food place close to the Café Rouge.* |
| name="Wildwood"; | eatType=restaurant; | familyFriendly=yes; | *Wildwood is a restaurant. Great for kids.* |
| name="The Punter"; | food=French; | priceRange=cheap; | *The Punter is a cheap French place.* |

$$D_{train} = \{mr^1 : x^1, \dots, mr^M : x^M\}$$

$$\mathrm{mr} = (a_1 : v_1, \dots, a_r : v_r)$$

$$\mathrm{x} = [x_1, \dots, x_N]$$

***V*** *is the vocabulary of references.*

*We want to learn a model that can generate sentences, given an input meaning representation!*

# Rule-based NLG

name="San Jalisco";
food=Mexican;
near=river;

Simplest method would be to create some templates!

- Hand-written lexicons, rules, templates, and schemata.

***[?name]*** *[is near the]* ***[?near]*** *[, and serves]* ***[?food]*** *[food.]*

***San Jalisco*** *is near the* ***river****, and serves* ***Mexican*** *food.*

# Rule-based NLG

name="San Jalisco";
food=Mexican;
near=river;

Simplest method would be to create some templates!

- Hand-written lexicons, rules, templates, and schemata.

***[?name]*** *[is near the]* ***[?near]*** *[, and serves]* ***[?food]*** *[food.]*

***San Jalisco*** *is near the* ***river****, and serves* ***Mexican*** *food.*

☻ High quality texts! Most industrial applications use templates.

☹ Time consuming to create.

☹ Requires expert knowledge.

☹ Domain-dependent.

# Retrieval-based NLG

| | | | |
|---|---|---|---|
| name="Cotto"; | food=Fast food; | near="Café Rouge"; | *Cotto is a Fast food place close to the Café Rouge.* |
| name="Wildwood"; | eatType=restaurant; | familyFriendly=yes; | *Wildwood is a restaurant. Great for kids.* |
| name="The Punter"; | food=French; | priceRange=cheap; | *The Punter is a cheap French place.* |
| name="The Vaults"; | eatType=restaurant; | food=Chinese; | *The Vaults is a Chinese restaurant.* |
| name="The Eagle"; | food=Fast food; | customerRating=low; | *The Eagle is a low rated fast food.* |

name="San Jalisco";
food=Mexican;
near=river;

| | | |
|---|---|---|
| name="Cotto"; | food=Fast food; | near="Café Rouge"; |

***[?name]*** *[is a]* ***[?food]*** *[place close to the]* ***[?near]*** *[.]*

# Retrieval-based NLG

| | | | |
|---|---|---|---|
| name="Cotto"; | food=Fast food; | near="Café Rouge"; | *Cotto is a Fast food place close to the Café Rouge.* |
| name="Wildwood"; | eatType=restaurant; | familyFriendly=yes; | *Wildwood is a restaurant. Great for kids.* |
| name="The Punter"; | food=French; | priceRange=cheap; | *The Punter is a cheap French place.* |
| name="The Vaults"; | eatType=restaurant; | food=Chinese; | *The Vaults is a Chinese restaurant.* |
| name="The Eagle"; | food=Fast food; | customerRating=low; | *The Eagle is a low rated fast food.* |

name="San Jalisco";
food=Mexican;
near=river;

| | | |
|---|---|---|
| name="Cotto"; | food=Fast food; | near="Café Rouge"; |

***[?name]*** *[is a]* ***[?food]*** *[place close to the]* ***[?near]*** *[.]*

However:

- Not all references may be fully delexicalizable.
- We want our model to generalize to unseen data!

name="The Place";
food=French;
familyFriendly=yes;

# ML-based NLG

First we preprocess our data: delexicalize values when able.

| | | | |
|---|---|---|---|
| name="Cotto"; | food=Fast food; | near="Café Rouge"; | ***?name*** *is a* ***?food*** *place close to the* ***?near*** *.* |
| name="Wildwood"; | eatType=restaurant; | familyFriendly=yes; | ***?name*** *is a* ***?eatType*** *. great for kids .* |

Gold NL sentences need to be aligned with MR concepts

- May be manually annotated (expensive/time-consuming).
- But automatic alignment methods also exist!

| | | |
|---|---|---|
| name="Wildwood"; | eatType=restaurant; | familyFriendly=yes; |

***?name*** *is a* ***?eatType*** *. great for kids .*

***name*** ➡ ***eatType*** ➡ ***familyFriendly***

# Hierarchical Language Models

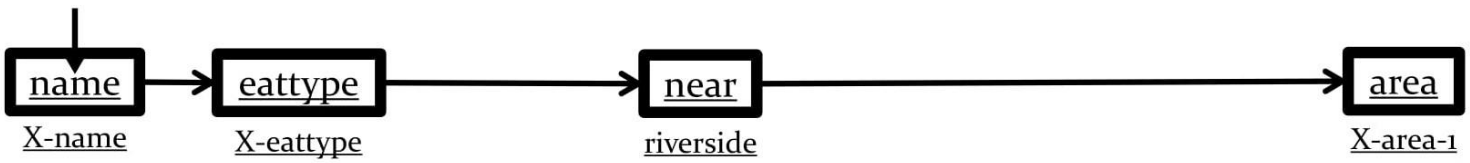
To generate the sentence we can combine language models:

- First, use a LM trained on concept sequences.

$$P(a) = \prod_{l=1}^{L} P(a_l | a_{l-1}, \dots, a_{l-k})$$

- And then, for each concept $a_l$, use a LM trained on the words aligned to $a_l$.

$$P_{a_l}(x) = \prod_{n=1}^{N} P(x_n | x_{n-1}, \dots, x_{n-k})$$



# Hierarchical Language Models

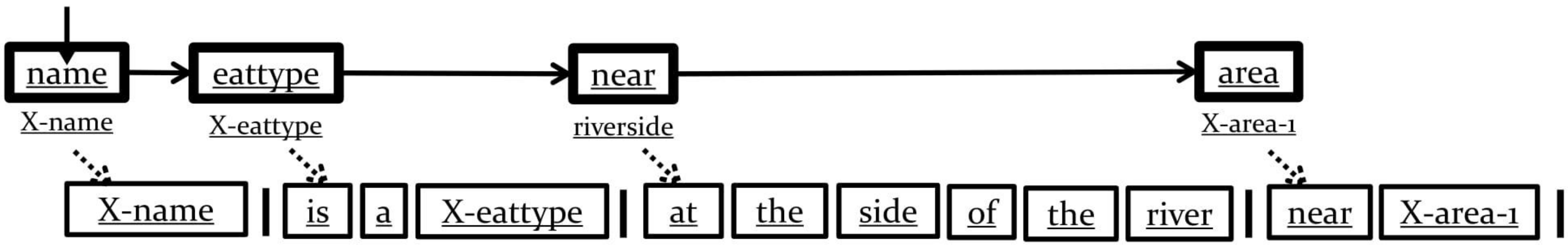
To generate the sentence we can combine language models:

- First, use a LM trained on concept sequences.

$$P(a) = \prod_{l=1}^{L} P(a_l | a_{l-1}, \ldots, a_{l-k})$$

- And then, for each concept $a_l$, use a LM trained on the words aligned to $a_l$.

$$P_{a_l}(x) = \prod_{n=1}^{N} P(x_n | x_{n-1}, \ldots, x_{n-k})$$



# Evaluation

Automatic evaluation:

- Compare against corpus-based gold standards, using automatic measures (e.g. BLEU, NIST, ROUGE).

BLEU, modified n-gram precision metric:

- To evaluate word choice, we want to use precision.

| | | |
|---|---|---|
| name="Cotto"; | food=Fast food; | near="Café Rouge"; |

Candidate generation #1

*is is is is is is is*

Unigram precision **P** = **7/7**

Gold reference

*Cotto is near the Café Rouge and is serving Fast food .*

Candidate generation #2

*Cotto is a Fast food place close to the Café Rouge .*

Unigram precision **P** = **9/11**

Gold reference

*Cotto is near the Café Rouge and is serving Fast food .*

Papineni et al., ACL 2002
"BLEU: a Method for Automatic Evaluation of Machine Translation"

# Evaluation

- Modified word precision considers the count of each word per reference.

$$p_n = \frac{\sum_{ngram \in C} Count_{clip}(ngram)}{\sum_{ngram \in C} Count(ngram)}$$

Candidate generation #1

*is is is is is is is*

Unigram mod. precision $\mathbf{p_1} = \mathbf{2/7}$

Gold reference

*Cotto is near the Café Rouge and is serving Fast food .*

max(is) = 2

Candidate generation #2

*Cotto is a Fast food place near the Café Rouge .*

Unigram mod. precision $\mathbf{p_1} = \mathbf{9/11}$

Gold reference

*Cotto is near the Café Rouge and is serving Fast food .*

# Evaluation

- To evaluate word order, we use bigger n-grams than unigrams (usually N=4).
  - BLEU uses the average logarithm of precisions with uniform weights, because the n-gram precision decays exponentially with n.

$$\sum_{n=1}^{N} w_n \, log p_n$$

Candidate generation #2

*Cotto is a Fast food place near the Café Rouge .*

Gold reference

*Cotto is near the Café Rouge and is serving Fast food .*

**$p_1 = 9/11$**
**$p_2 = 5/10$**
**$p_3 = 2/9$**
**$p_4 = 1/8$**

$$\frac{1}{4} log p_1 + \frac{1}{4} log p_2 + \frac{1}{4} log p_3 + \frac{1}{4} log p_4$$

Chen and Cherry, ACL 2014
"A Systematic Comparison of Smoothing Techniques for Sentence-Level BLEU"

# Evaluation

- Finally, to match the NL references in length, BLEU also uses a brevity penalty.

Candidate generation #3

*Cotto is near the Café Rouge*

Gold reference

*Cotto is near the Café Rouge and is serving Fast food .*

**$p_1$ = 6/6**
**$p_2$ = 5/5**
**$p_3$ = 4/4**
**$p_4$ = 3/3**

c=6

r=12

- The brevity penalty only occurs if the candidate is shorter than any of the references.

$$BP = \begin{cases} 1 & if\ c > r \\ e^{\left(1-\frac{r}{c}\right)} & if\ c \leq r \end{cases}$$

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \, log p_n\right)$$

$$BP = e^{\left(1-\frac{12}{6}\right)} = e^{-1} \approx 0.37$$

$$BLEU = 0.37 \cdot exp(0) = 0.37$$

# Human evaluation

Automatic evaluation can be problematic.

| | |
|---|---|
| *The Fast food place Cotto is near Café Rouge .* | *Cotto is near Café Rouge and is serving Fast food .* |

We can ask human judges to evaluate the output of the NLG system.

- Most often asked to score a variety of criteria on a Likert scale (1-5) :

  Fluency/grammaticality/naturalness
  i.e. how probable is that a human would write this sentence?

  Meaning preservation / informativeness
  i.e. does the sentence express all the information in the meaning representation?

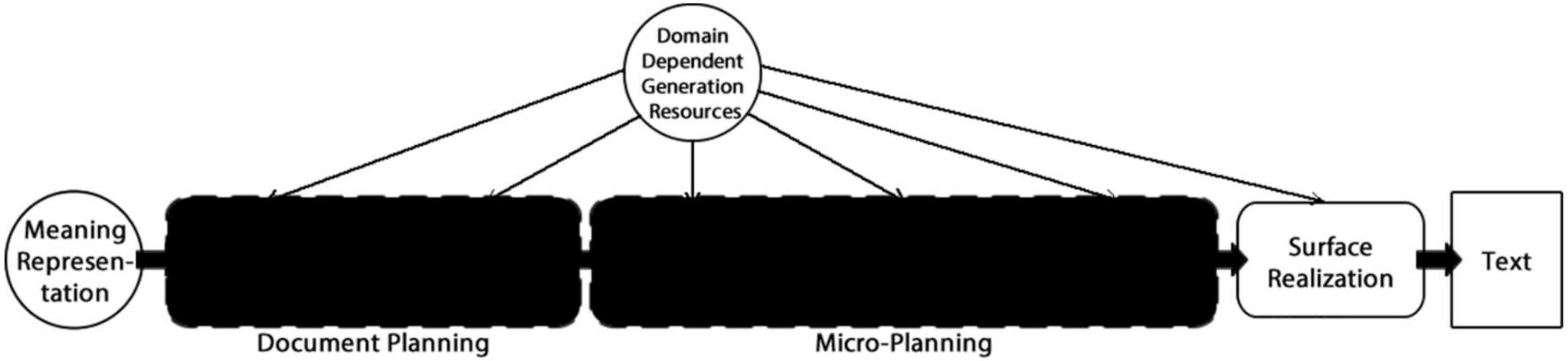
Human evaluation has its own drawbacks:

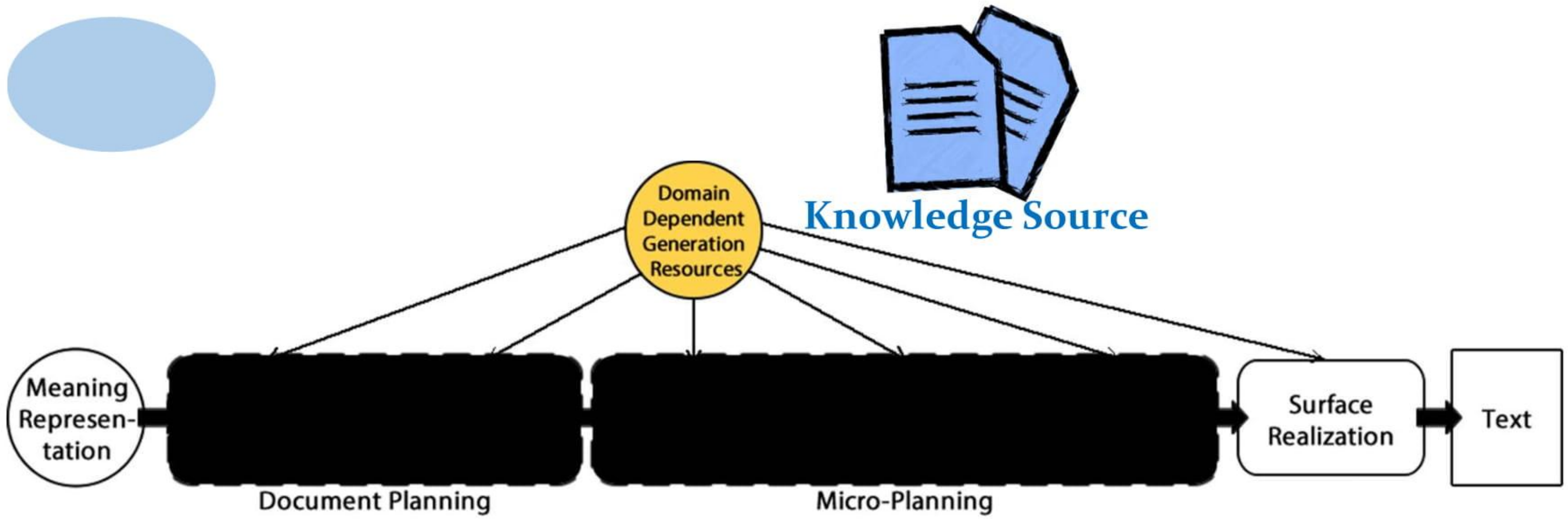
- It can be very expensive.
- Requires careful training and filtering of unreliable judges.
- In some cases it can be subjective.

# NLG stages. So many choices!

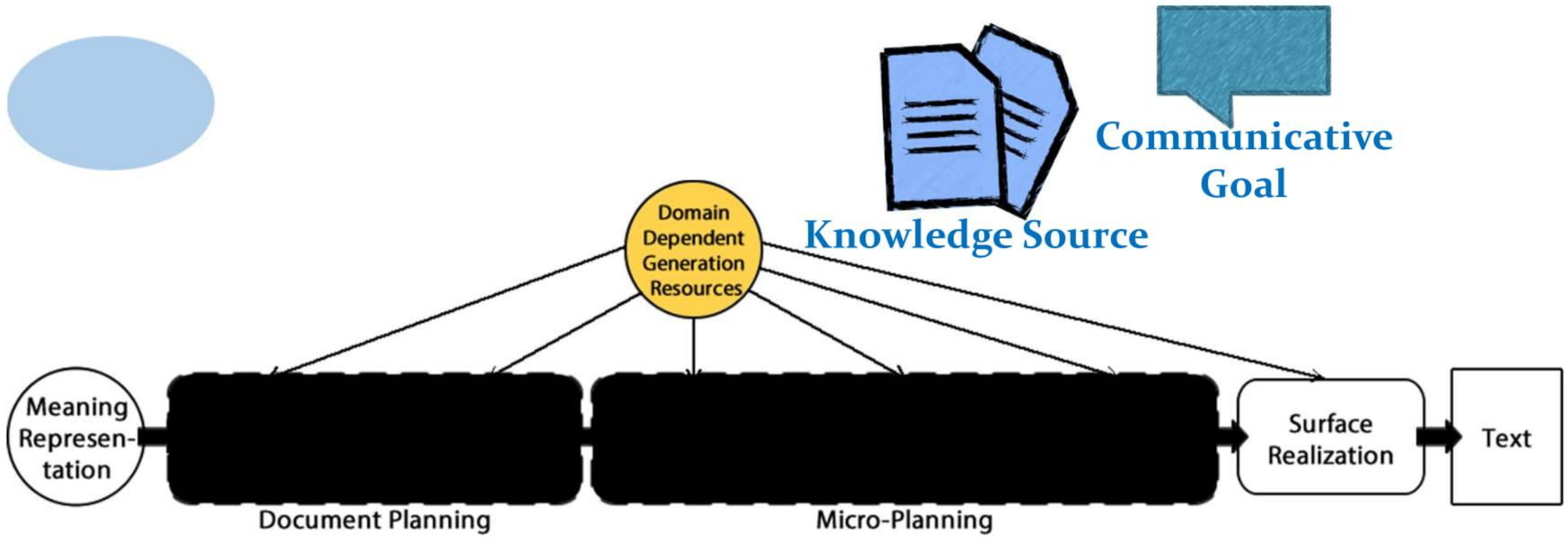
We have discussed models that mostly address how to express concepts with words, but there are more choices.

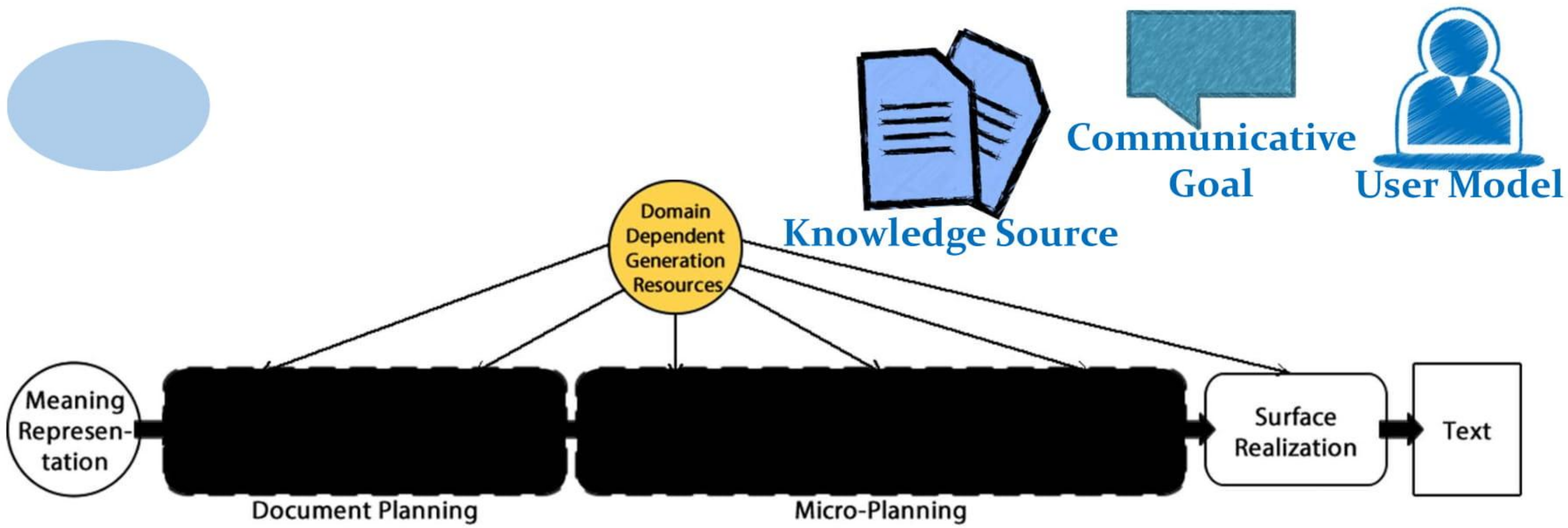
- What to say?
- How to say it?
- What is everything called?
- How to structure it?
- Who am I?
- Who am I talking to?


Domain
Dependent
Generation
Resources
Meaning
Represen-
tation
[redacted]
[redacted]
Surface
Realization
Text
Document Planning
Micro-Planning

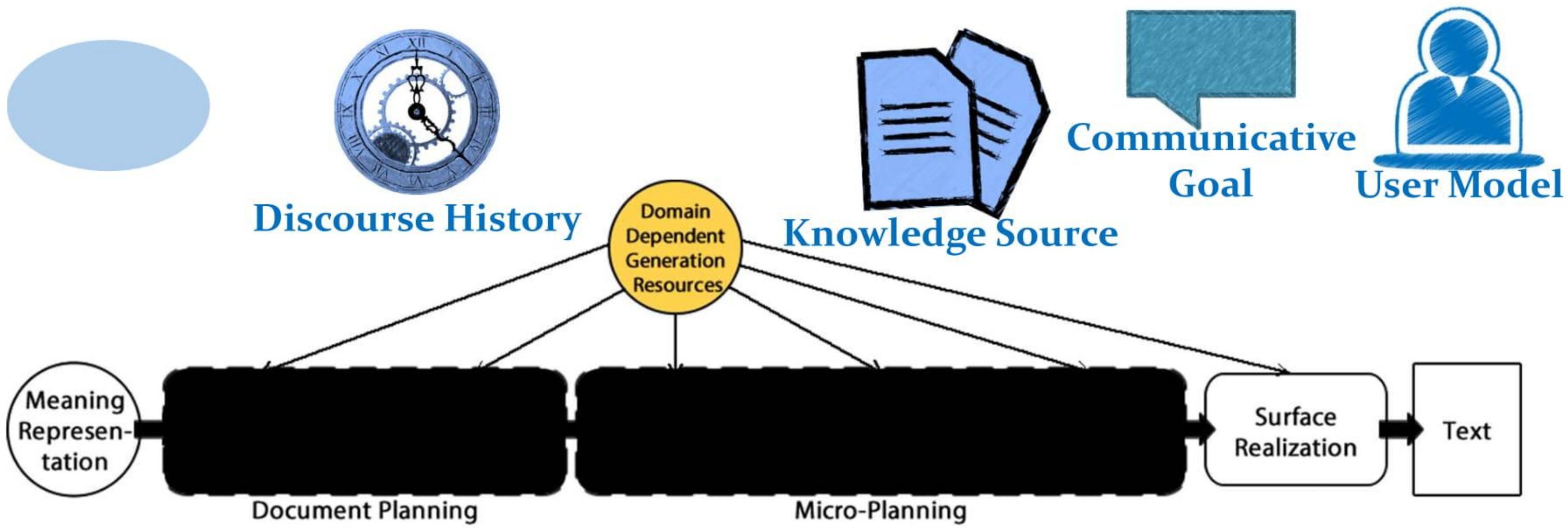



- Domain Dependent Generation Resources
  - The content and representation highly depends on the application.
  - Generally rule-based or ML-based models trained on data.


Communicative
Goal
Knowledge Source
Domain
Dependent
Generation
Resources
Meaning
Represen-
tation
[redacted]
Document Planning
[redacted]
Micro-Planning
Surface
Realization
Text

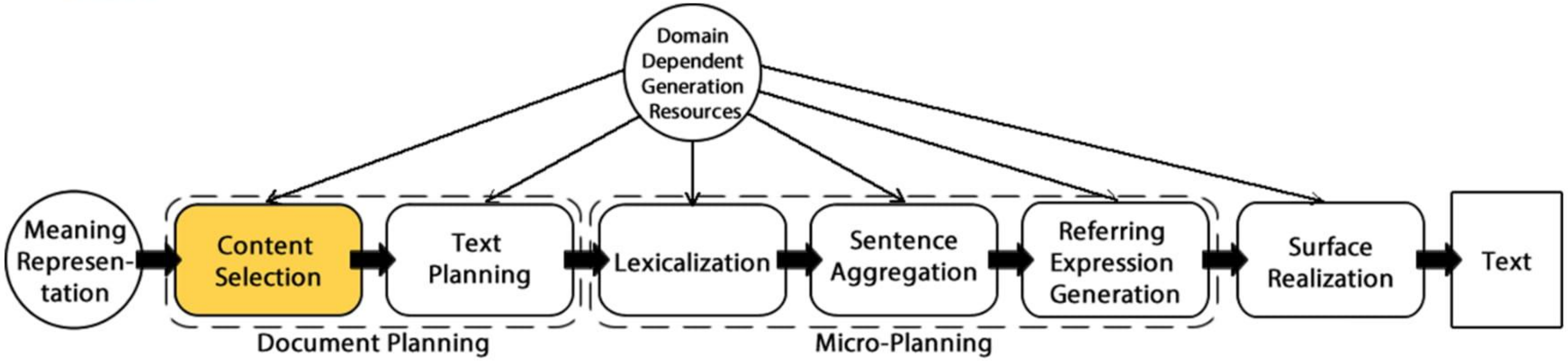



- Domain Dependent Generation Resources
  - User model
    - Information about the intended audience of the text.
    - Implicit user model:
      - Hard wired, no adaption to different types of users.
    - Explicit user models:
      - Hard wired variations of users to account for expected stereotypes, i.e. child, adult, expert
    - Personal user model, adaptable through user interactions.




- Domain Dependent Generation Resources
  - Discourse history
    - Record of what has already been communicated to the user.
    - Relevant when generating texts in sequence (e.g. in a dialogue).

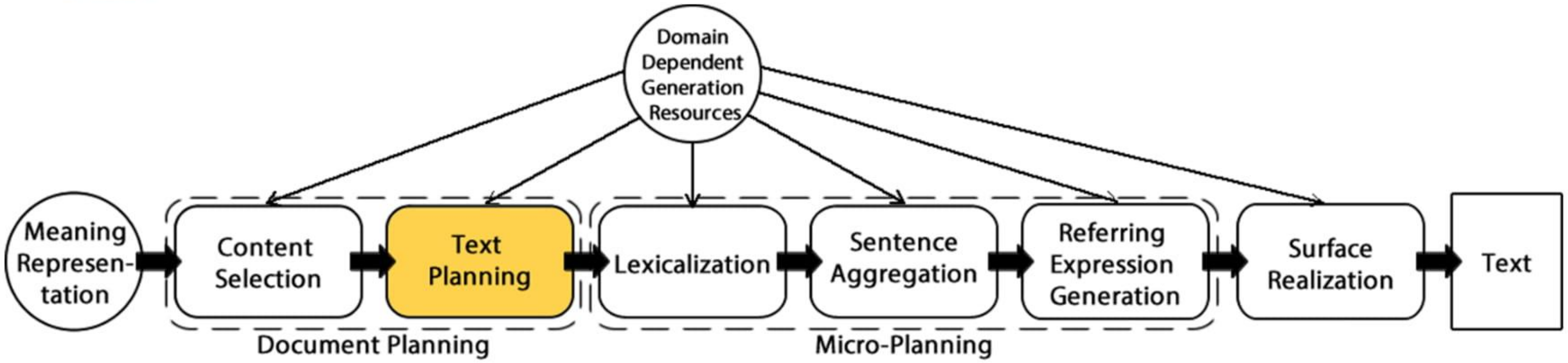
    - Can affect pronoun generation, content selection, rhetorical structure, and comparison generation.




*:StEmilion, isA, :Bordeaux*
*:StEmilion, :hasColor, :red*
*:StEmilion, :locatedIn, :stEmilionRegion*
*:StEmilion, :madeFrom, :cabernetSauvignonGrape*

Select which information to convey in the text.

- Space may be limited.
- User model may indicate content preferences.

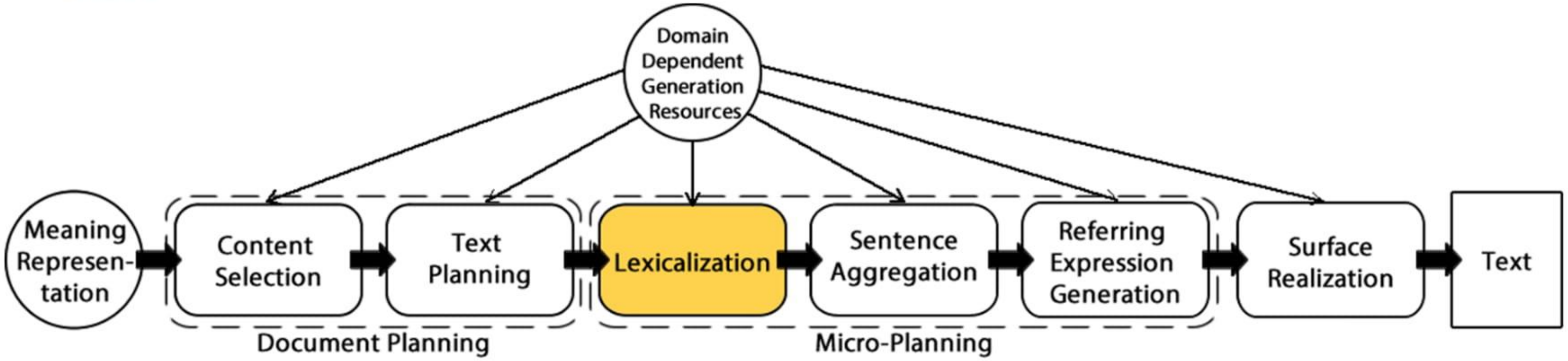



**Description section**

*:StEmilion, isA, :Bordeaux*
*:StEmilion, :hasColor, :red*

**Production section**

*:StEmilion, :madeFrom, :cabernetSauvignonGrape*



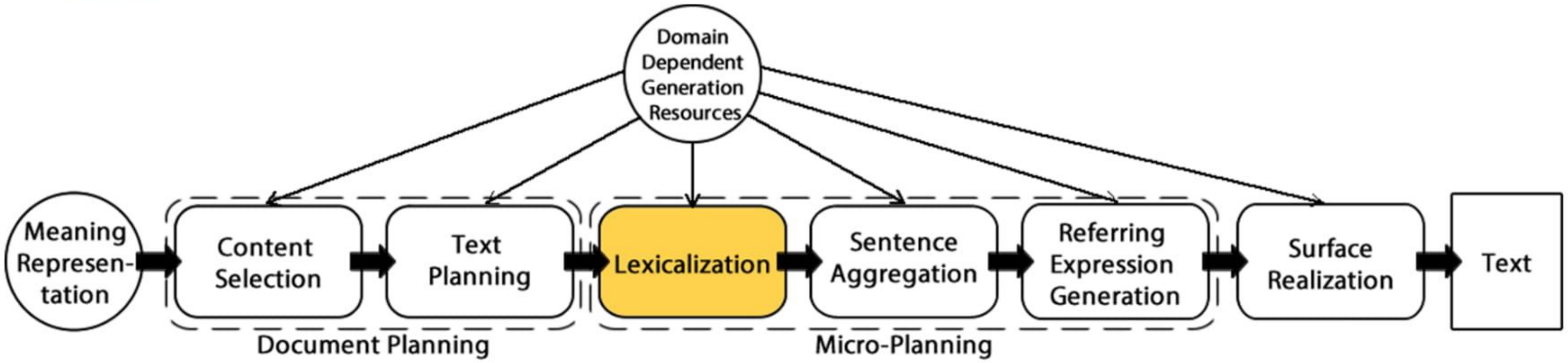

*:StEmilion, isA, :Bordeaux*
*:StEmilion, :hasColor, :red*

→

*[:StEmilion] [toBe] [a kind of] [:Bordeaux]*
*[:StEmilion] [toBe] [:red]*

Words/syntactic structures to express each message.

- St. Emilion is a kind of Bordeaux.
- St. Emilion is a Bordeaux wine.
- If you are looking for a Bordeaux wine, try St. Emilion.
- The St. Emilion wine can be classified as a Bordeaux.



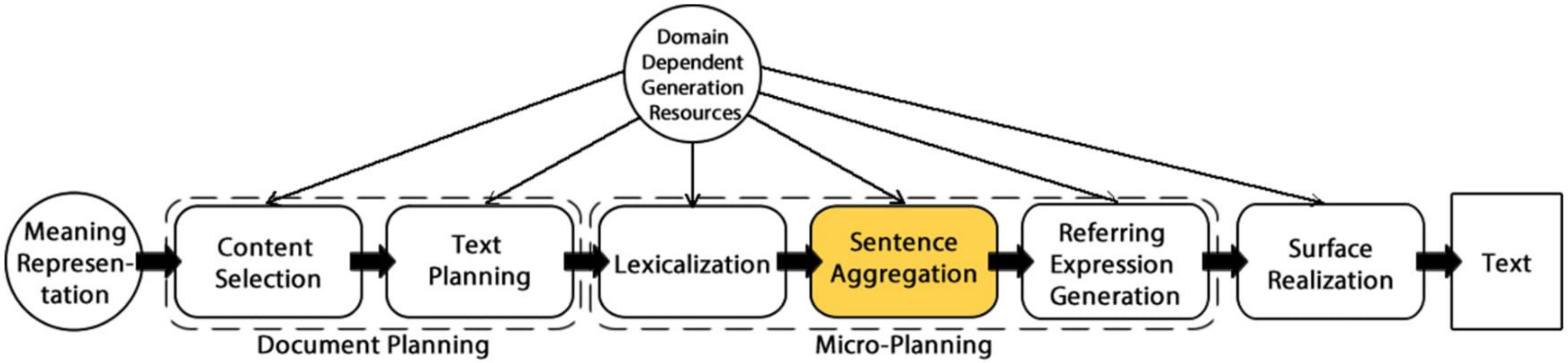

*:StEmilion, isA, :Bordeaux*
*:StEmilion, :hasColor, :red*

→

*[:StEmilion] [toBe] [a kind of] [:Bordeaux]*
*[:StEmilion] [toBe] [:red]*

Need to consider:

- User model knowledge and preferences.
- Consistency with discourse history vs. variation in text.
- Interactions with other NLG stages.

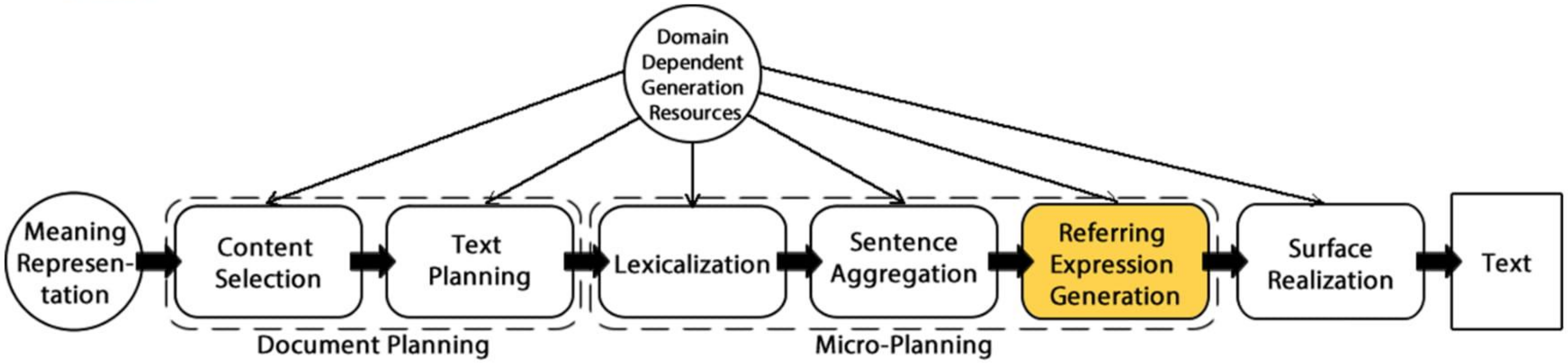



*[:StEmilion] [toBe] [a kind of] [:Bordeaux]*
*[:StEmilion] [toBe] [:red]*

*[:StEmilion] [toBe] [a kind of] [:red] [:Bordeaux]*

## When and how should we form sentences?

- Which messages should be combined?
  - Information content / Semantics.
- How long should sentences be?



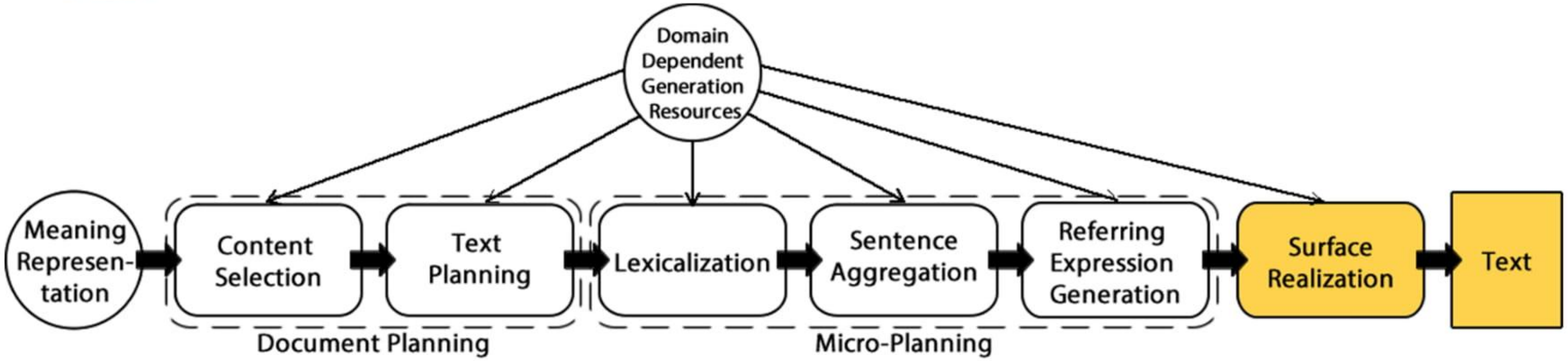

*[:StEmilion] [toBe] [a kind of] [:red] [:Bordeaux]*

⇩

*[This] [toBe] [a kind of] [red] [Bordeaux]*

Refer to an entity using its name, a pronoun, or a [in]definite noun phrase?

- Discourse history for pronouns.
  - As long as they are not ambiguous.
- Elaborate linguistic theories on this…



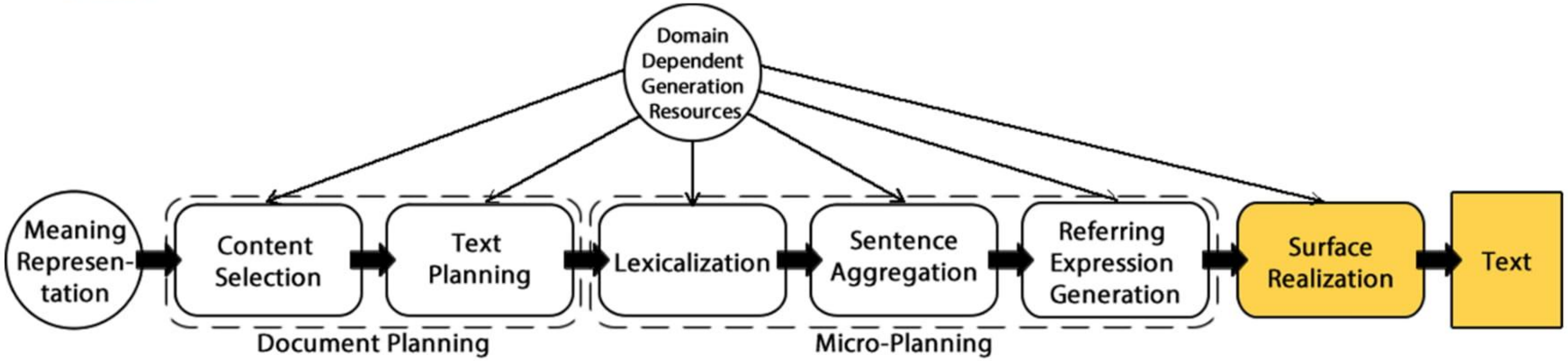

*St. Emilion is a red strong Bordeaux.*

*It is made from exactly one grape variety: Cabernet Sauvignon grapes.*

Generates texts / final details:

- Ensure syntactic / morphological agreement, orthographical correctness.

Peculiarities of language shouldn't affect previous stages.




- We've seen models that can generate coherent text!

- The models we discussed addressed:
  - text planning (concept ordering)
  - lexicalization
  - surface realization

- While there are rule-based and ML-based approaches for all these stages.
  - Some models consider the decisions across stages jointly.

# Bibliography

Building Natural Language Generation Systems,
Ehud Reiter and Robert Dale,
Cambridge University Press, 2006.

A recent NLG survey
by Albert Gatt and Emiel Krahmer.

An NLG overview
by Ehud Reiter and Robert Dale.

Presentations
of a (not so) recent Summer School on NLG.