Natural Language Generation: Traditional Approaches and Research Directions

Lecture 2: Statistical Approaches

Pablo Ariel Duboue, PhD.

Textualization Software Ltd.
Vancouver, Canada

17th Estonian Summer School on Computer and Systems Science

Outline

1. Statistical NLG
   - Data Acquisition
   - Evaluation
2. Methods
   - General Statistics
   - Language Specific
   - NLG Specific
3. Examples
   - Selected Papers
   - ProGenIE

Review from Lecture 1

- NLG at its core deals with representing and handling large number of principled decisions.
  - A process of enriching the input representation culminating into a full fledged text.
- Not necessarily the inverse of NLU.
  - Deals with communicative intention much more than NLU.

Review from Lecture 1

- NLG Subtasks:
  - Content Planning.
    - Content Selection.
    - Document Structuring.
  - Sentence Planning.
    - Aggregation.
    - Referring Expression Generation.
    - Lexicalization.
  - Surface realization.
    - Linearization.

Feedback from yesterday

Overall very positive. Thanks! These are questions and feedback that deserved response but it is not representative of the rest.

- Go slower. Explain figures in more details. I was unable to follow the code examples. *My mistake, I got lost with the end time of my lecture given the photo break.*
- How to generate flousihes without previously overspecifing the world?
  - "Mary sat on a couch" vs. "Mary sat on the old leather couch"
  - *If it is not in the input it is computational creativity, if you go there make sure you tell your readers.*
- Is LISP still in use in the fields of NLP/NLG? *No, but Haskell is. My current work is in scala.*
- Loglan/Lojban projects – are they related to the NLG problems? *I had hoped to see more of this but not at the moment.*

Feedback from yesterday

- What do you think about Generative Grammar theory? *CFGs are very used in terms of standard theory. I like LFG and we'll see an example today of TAG (Tree Adjoining Grammars).*
- Examples of what to say in some abstract representation. *We will see ProGenIE today.*
- How NLP-based systems understand the priority/confidence score in a sentence? *Sounds like an interesting question, but I don't undersand it, come see me during the break.*
- I'd like to know more about the details of your algorithms so I can use them in my work. *Nice! Talk to me during the break.*
- I'd like to hear more about theory rather than tools.
    - *Theory means different things for different people. For NLG, subtasks and their ordering are theoretical discussions. I prefer examples rather than theory to reach to a wide audience. For yesterday, most theory requires linguistic beyond CompSci.*

Today

- Statistical methods
- Data acquisition
- Evaluation
- ProGenIE: Profile Generation by Information Extraction

Outline

Data-Text Corpora

- Acquiring Data-Text corpora is difficult.
  - Writers do not need the type of data a NLG system needs.
    - Plenty of text, lack of data.
  - When data is available, writing is truly an issue.
    - Plenty of data, few text examples, written for that purpose.
    - Expensive.
    - Non-expert writters.
- And “data” is very ambiguous.
  - Data does not necessarily mean “data we can use to generate the text we want”.

Poor Quality Text

- Also, in NLU we want to make the system robust.
  - Perform well under a variety of inputs (texts).
- To do the same in NLG we should make sure it handles data with mistakes.
  - A seldom investigated topic.
    - On which I have centered my efforts in recent years.
- But training NLG with large amounts of naturally occurring text means we are training on poor outputs.
  - That is undesirable, we want to generate the best possible text.

Training Subtasks

- Finding Data-Text training data is already hard.
- Finding training data involing input-output pairs for a specific subcomponent is very, very rare.
- For this purpose people either :
  - Transform available NLU training data.
    - But the transformation process might do too much work and render the learning contribution unclear.
  - Migrate or align the data-text pair into input-outputs for the subtask.
    - An approach I employed on my thesis and we will talk today.

Outline

Evaluation

- Intrinsic:
  - Subjective.
    - Readability.
    - Grammaticality.
    - Appropriateness.
  - Corpus-based.
    - BLEU.
    - ROUGE.
- Extrinsic.

Subjective: What to Ask

- 5 point Likert scale.
- Slider (floating point).
- Comparing two outputs.
- Compare to a “modulus”.
  - Magnitude estimation used by Siddharthan & Katsos, 2012.
- Belz & Kow, 2010 used a preference-based paradigm.
  - Found it more sensitive to differences between systems.
  - Less sensitive to differences between people.

Subjective: Inter-rater Agreement

- Kappa Statistic and variants.
- High variance, for example in Question Generation, Rus et al, 2011.
- Iterative updating of guidelines with discussion helps reduce variance, Godwin & Piwek, 2016.
- Amazon Mechanical Turk.
  - Ethical issues.

Subjective: Readability

- Also known as fluency.
- Ask the raters whether the text is readable, fluent, easy to understand.
    - Different from whether the text is well-formed or useful.

Subjective: Grammaticality

- Whether the text is correct to prescriptive grammar guidelines.
  - As understood by the raters.
- A text might be grammatical and very difficult to understand.
- Some people have very little tolerance to grammatical errors.
  - They would make good raters for this metric.

Subjective: Appropriateness

- Also known as accuracy, adequacy, relevance or correctness relative to the input.
  - Reflects content selection choices.
- In my experience, this metric is key.
  - I did my PhD in Content Selection so I am very biased.
  - People will tolerate poor text inasmuch the content is there.
    - On the other hand, great prose without the needed information is not very useful.

Corpus-based

- Corpus-based metrics have the advantage of being easy (and cheap) to reproduce.
- Three major types:
  - *n*-gram overlaps.
    - Used for evaluating surface realizers or short texts e.g., weather reports or captioning.
  - Edit distance.
    - Used in realization and REG.
  - Information overlap.
- Corpus-based metrics focus only on the output text
  - Nothing to do with the input.
  - There are some counter examples, Reiter & Belz, 2009 or Banik et al 2013.

Information Retrieval Metrics

- Measuring how many times a system outputs the right answer ("accuracy") is not enough.
  - Many interesting problems are very biased towards a background class.
  - If 95% of the time something doesn't happen, saying it'll never happen (not a very useful classifier!) will make you only 5% wrong.
- Metrics:

$$precision = \frac{|correctly\ tagged|}{|tagged|} = \frac{tp}{tp + fp}$$

$$recall = \frac{|correctly\ tagged|}{|should\ be\ tagged|} = \frac{tp}{tp + fn}$$

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

BLEU

- Comes from Machine Translation.
  - Defined in Papineni et al, 2002.
  - Precision over variable length *n*-grams, with a length penalty.
- For Machine Translation, it has been shown it correlates with human judgments.
  - For the quality of MT texts produced in 2002.
  - Having multiple reference translations was key to its success.

ROUGE

- Comes from summarization.
  - Defined in Lin & Hovy, 2003.
  - Recall oriented for comparing non-contiguous *n*-grams and longest common subsequences.
    - Length is already fixed because it is used in summarization.
- Many, many variants.
  - My students described as a “shotgun approach to evaluation”.

Other Corpus-based Metrics

- Meteor, from MT.
  - Harmonic mean of unigram precision and recall with options for handling (near-synonymy) and stemming, Lavie & Agarwal, 2007.
- CIDEr, from image captioning.
  - Cosine-based *n*-gram similarity score, with *n*-gram weighting using TF-IDF, Vedantam et al, 2015.
- WMD word-mover distance, from document similarity / image captioning.
  - Using semantic distance between words in the texts, where semantic uses word embeddings from Mikolov et al, 2013.
  - Presented in Kusner et al, 2015.
- Based on edit distance (Levenshtein, ...).
- Based content overlap (Jaccard, ...).

Corpus-based vs. Subjective

- Many times they do not correlate, see Gatt & Belz, 2010.
- Sometimes a system does not outperform on BLEU but humans find it better strongly, see Kiros et al, 2014.
- For comparison between metrics in image captioning, see Elliott & Keller, 2014.
    - In that domain, it seems Meteor is more robust.
- See Reiter & Belz, 2009 for discussion.

Extrinsic: GiVE Challenge

- The Giving Instructions in Virtual Environments was a 3D dungeon instruction-giving evaluation run in 2011.
- Besides expensive, they are very hard to replicate.
- Note how fluency or grammaticality sometimes has nothing to do with extrinsic metrics.

Black Box vs. Glass Box

- Most evaluation is black box end-to-end.
  - Difficult to have raters analyze the output of intermediate components.
- There are cases of glass-box.
  - Callaway & Lester, 2002), were ablation allowed measuring the impact of different features.
- In my own experiments, I evaluated against an automatically reconstructed output from human texts.
  - This evaluation penalizes my approach, though as my answers might have been correct albeit different from the reference text.
  - Similar problem with end-to-end using only one reference text.

Evaluation Wrap-up

- Need multiple evaluation metrics.
- Meta-evaluations show it is a toss-up which metric works better.
  - The genre seems to influence whether corpus and subjective correlate.
  - Ongoing research.
- Receiver-oriented metrics (how the text is processed) are under-explored.

Outline

Methods

- General Statistical:
  - Learning Orderings.
  - EM Algorithm.
  - Viterbi Decoding.
- Language Specific:
  - Language Models.
- NLG Specific:
  - Generate-and-Rank.
    - Over-generating Grammars.
  - Alignment.

Learning Orderings: Problem

- Input: example sequences of elements
- Output: total order of said elements
- Issues
  - Which elements conclusively should appear before each other?
  - Dealing with noise.
- Example: { A, C, D }; { A, B, D } ; { B, A, C } ; { A, D, C } ; { A, B, C }; { C, B, D }

Learning Orderings: Hypothesis Testing

- To induce a total order, build a table of occurrence counts for each possible pair of elements.
- The count is the number of times the element in the row appeared before the elements in the column.
- From this table it is possible to perform a statistical test
  - Determine whether we can reject the null hypothesis that the element in the row is statistically likely to come before the element in the column.
- Example

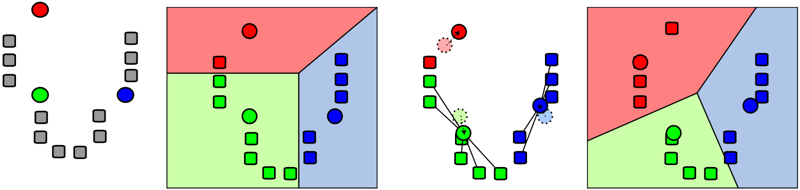
	A	B	C	D
A		2	3	3
B	1		2	1
C		1		2
D			1	

Learning Orderings: Topological Sort

- The pairs whose order is statistically significant form a lattice (a tree).
- A total order can be read out by using topological sort.
- At each stage, we remove a leaf from the tree.
  - Pick one at random if there are multiple leaves available.
- The resulting sequence defines a total order

EM Algorithm

- Expectation-Maximization is a classic algorithm to train statistical systems.
- Slow to converge.
- Two steps:
  - Fix model, estimate parameters (E-step).
  - Fix parameters, estimate model (M-step).
- Example, clustering using k-Means:



(Wikipedia)

Viterbi Decoding

- Efficient way to read the most likely path on a directed graph annotated with probabilities.
  - For example, a sentence from a packed forest.
- Dynamic programming algorithm.
  - Works due to limited memory assumptions.

Outline

Language Models

- A language model is a probability model that describes what strings are more likely in a given language.
  - "The dog is" is a likely string.
  - "is The dgo" is a very unlikely string.
- Simple models use the probability of two consecutive words.
  - Used in predictive keyboards as found on cellphones.
- Key component of speech recognition.
  - Compare (from Wikipedia):
    - *How to recognize speech using common sense.*
    - *How to wreck a beach using calm incense.*

Over-generating Grammars

- Write (usually by hand) a map from the input to the NLG system to small (P)CFG grammars.
  - Will generate correct language.
  - Also grammatically incorrect or unusual language.
- This reduces the human effort needed to write a generator.
- The grammars can also be estimated:
  - From a corpus of grammatically annotated text.
    - Tree bank.
  - From the output of parser run over a text corpus.

Over-generating Grammars: Example

- These are some tables from a system generating random text.
  - `http://firstsentence.net`
  - Trained on 60k first paragraph sentences of project Gutenberg.

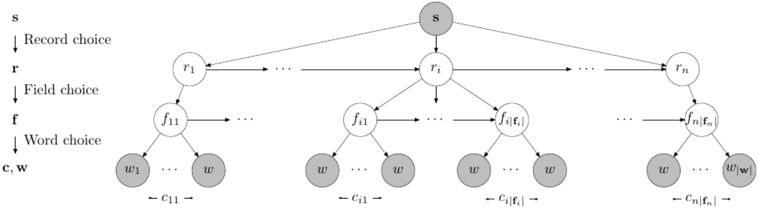
'S' =>
'NP, NP VP.'/139,
'SBAR, NP ADVP VP.'/94
'RB PP NP VP' /8,
'S, CC S CC S'/3,...

'NP' =>
'DT NN'/84638,
'NP JJ JJ'/2,
'RB DT ADJP NNS CC NNS'/1, ...

'NNS' => 'ships'/108, 'acquaintances'/33, 'seeds'/22, 'alleys'/6, 'yew-trees'/1, 'buggies'/1 , ...
'WP' => 'what'/1804, 'who'/4295, 'whom'/699, 'whoever'/ 15, ...
'VBD' => 'prevented'/34, 'breathed'/22, 'lowered'/14, 'dined'/13, 'elaborated'/1, 'patronised'/1, ...

Aligment

- From Liang et al, 2009



Outline

Generate-and-Rank

- Using a simple grammar or another method (like a reversible NLU component), generate multiple alternative outputs.
  - For example, multiple sentence plans.
- Still, the central issue in NLG is one of choice.
  - Still remains to choose among those outputs.
- Choose by leveraging a language model or any other metric of "good output".
  - Separate the creation from the scoring.
  - The scoring is independent of generation.
    - How similar the output is to the target text as a whole.

Outline

1. Statistical NLG
   - Data Acquisition
   - Evaluation

2. Methods
   - General Statistics
   - Language Specific
   - NLG Specific

3. Examples
   - Selected Papers
   - ProGenIE

Shaw & Hatzivassiloglou

- *"Learning ordering among premodifiers"*, ACL-99.
  - As part of the MAGIC project.
- Why *"a 21-year-old Caucasian male patient of Dr Smith"* and not *"Dr. Smith's male Caucasian 21-year-old patient"*?
- Approach:
  - Collect a corpus of target expressions.
  - Transform them into sequences of semantic types (<age, race, gender, dr>).
  - Use the ordering learning algorithm to extract a total order.
  - Generate using the total order.

Langkilde & Knight

- *"The practical value of n-grams in generation"*, INLG-98.
- Generating English from a Japanese input sentence.
  - Articles are an educated guess, at most.
  - Generate-and-rank does precisely that.
- Small grammar written by hand that overgenerates.
  - *n*-gram language model to pick the best output.

Langkilde & Knight: Example

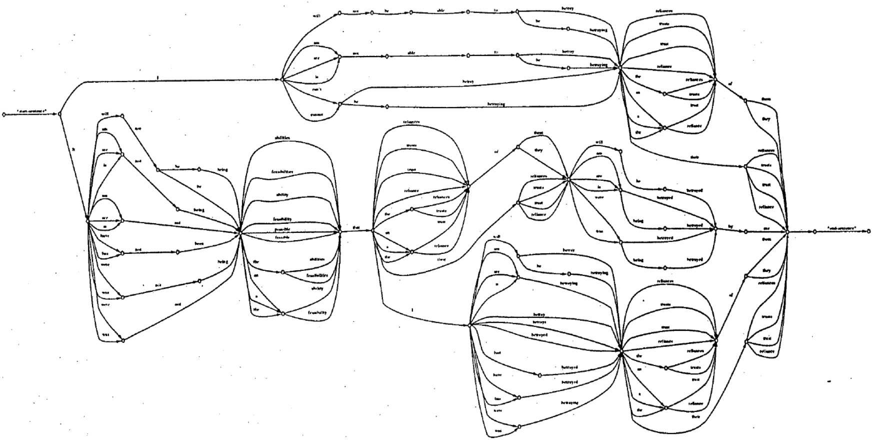
- Input: Augmented Meaning Representations (AMRs):

```
(A / |workable|
  :DOMAIN (A2 / |sell<cozen|
  :AGENT I
  :PATIENT (T / |trust,reliance|
               :GPI THEY))
  :POLARITY NEGATIVE)
```

- Word lattice has 270 nodes, 592 arcs, and 155,764 paths.
- Top paths:
  - *I cannot betray their trust .*
  - *I will not able be able to betray their trust .*
  - *I am not able to betray their trust .*
  - *I are not able to betray their trust .*
  - *I is not able to betray their trust .*
  - *I cannot betray the trust of them .*
  - *I cannot betray a trust of them .*

Langkilde & Knight: Lattice

- One-fifth of the sentence lattice



NLG as a Sequential Stochastic Process

- Angeli, G., Liang, P., & Klein, D. *A Simple Domain-Independent Probabilistic Approach to Generation*. EMNLP-2010.
- End-to-end generation.
  - Integrated content selection and surface realization.
  - Sequential local decisions trained discriminatively.
- Three domains:
  - RoboCup (robot soccer simulator).
  - Technical weather reports.
  - Common weather reports.
- Input is set of DB records with fields.
- Output is a sentence.

Angeli et al 2010: Task

s:

pass(arg1=`purple6`, arg2=`purple3`)
kick(arg1=`purple3`)
badPass(arg1=`purple3`,arg2=`pink9`)
turnover(arg1=`purple3`,arg2=`pink9`)

w: *purple3 made a bad pass that was picked off by pink9*

(a) ROBOCUP

Angeli et al 2010: Task

s:

temperature(time=5pm-6am,min=48,mean=53,max=61)
windSpeed(time=5pm-6am,min=3,mean=6,max=11,mode=0-10)
windDir(time=5pm-6am,mode=SSW)
gust(time=5pm-6am,min=0,mean=0,max=0)
skyCover(time=5pm-9pm,mode=0-25)
skyCover(time=2am-6am,mode=75-100)
precipPotential(time=5pm-6am,min=2,mean=14,max=20)
rainChance(time=5pm-6am,mode=someChance)

w: *a 20 percent chance of showers after midnight . increasing clouds , with a low around 48 southwest wind between 5 and 10 mph*

(b) WEATHERGOV

Angeli et al 2010: Task

s:

wind10m(time=6am,dir=SW,min=16,max=20,gust_min=0,gust_max=-)
wind10m(time=9pm,dir=SSW,min=28,max=32,gust_min=40,gust_max=-)
wind10m(time=12am,dir=-,min=24,max=28,gust_min=36,gust_max=-)

w: *sw 16 - 20 backing ssw 28 - 32 gusts 40 by mid evening easing 24 - 28 gusts 36 late evening*

(c) SumTime

Angeli et al 2010: Details

- Log-linear classifiers for each decision.
  - Domain independent features, can incorporate domain dependent, too.
  - Each decision d depends on the history of previous decisions:

$$p(d_j|\boldsymbol{d}_{<\boldsymbol{j}}, db; \theta) = \frac{\exp\{\phi_j(d_j, \boldsymbol{d}_{<\boldsymbol{j}}, db)^T\theta\}}{\sum_{d'_j \in \mathcal{D}} \exp\{\phi_j(d'_j, \boldsymbol{d}_{<\boldsymbol{j}}, db)^T\theta\}}$$

- Three classifiers:
  - Macro content selection (choose records from DB).
  - Micro content selection (choose fields from records).
  - Surface realization (choose template to verbalize fields).
- Generation stops when the "STOP" record is generated.
  - Generate r1, F1, T1, r2, F2, T2, ..., STOP and use a LM over the whole sequence.

Angeli et al 2010: Feature Templates

Record

- R1 list of last *k* record types
- R2 set of previous record types
- R3 record type already generated
- R4 field values
- R5 stop under language model (LM)

Field Set

- F1 field set
- F2 field values

Template

- W1 base/coarse generation template
- W2 field values
- W3 first word of template under LM

Angeli et al 2010: Feature Templates

Notation, $[\![e]\!] = 1$ iff expression e is true.

R1 list of last k record types
$[\![r_i.t = * \text{ and } (r_{i-1}.t, \ldots, r_{i-k}.t) = *]\!]$ for $k \in \{1, 2\}$

R2 set of previous record types
$[\![r_i.t = * \text{ and } \{r_j.t : j < i\} = *]\!]$

R3 record type already generated
$[\![r_i.t = r_j.t \text{ for some } j < i]\!]$

R4 field values
$[\![r_i.t = * \text{ and } r_i.v[f] = *]\!]$ for $f \in \textsc{Fields}(r_i.t)$

F1 field set $[\![F_i = *]\!]$

F2 field values $[\![F_i = * \text{ and } r_i.v[f] = *]\!]$ for $f \in F_i$

W2 field values
$[\![T_i = * \text{ and } r_i.v[f] = *]\!]$ for $f \in F_i$

Angeli et al 2010: Training

- Link the Data-to-Text using Liang et al., 2009 (which learns the mapping using EM).
  - Estimate the latent variable d as data is db, w is not a sequence of decisions d.
- Learn the weights θ using optimization (not unlike gradient descent).
  - Once the latent variables are estimated.
- Data sizes:
  - ~30,000 for WeatherGov.
  - ~1,000 for RoboCup.

Angeli et al 2010: Alignment

Records:	skyCover_1	temperature_1			
Fields:	mode=50-75		time=17-30	min=44	mean=49
Text:	*mostly cloudy ,*	*with a*	*low around*	*45*	*.*

⇒

Aligned training scenario

		skyCover	temperature
⇒	COARSE	$\langle$[mode]$\rangle$	$\langle$*with a* [time] [min] [mean]$\rangle$
	BASE	$\langle$*most cloudy ,*$\rangle$	$\langle$*with a low around* [min] *.*$\rangle$

Templates extracted

Angeli et al 2010: Generation

- They generate using a greedy decision process on the trained model.
  - Not much paraphrasing.
  - They can also sample from the probability distribution.
- Viterbi decoding is not possible on this model.
- Beam search performed worse than greedy.

Angeli et al 2010: Generation Example

World state

skyCover$_1$: skyCover(time=5pm-6am,mode=50-75)
temperature$_1$: temperature(time=5pm-6am,min=44,mean=49,max=60)
...

Decisions

Record	r_1 = skyCover$_1$	r_2 = temperature$_1$	r_3 = STOP
Field set	$F_1 = \{\text{mode}\}$	$F_2 = \{\text{time}, \text{min}\}$	
Template	$T_1 = \langle$*mostly cloudy ,*$\rangle$	$T_2 = \langle$*with a low around* [min] *.*$\rangle$	

Text *mostly cloudy , with a low around 45 .*

Angeli et al 2010: Generation Example

$$r_2 = \text{temperature}_1$$

$$
\begin{array}{ll}
\textbf{(R1)} & [\![r_2.t = \text{temperature and } (r_1.t, r_0.t) = (\text{skyCover}, \textsc{start})]\!] \\
 & [\![r_2.t = \text{temperature and } (r_1.t) = (\text{skyCover})]\!] \\
\textbf{(R2)} & [\![r_2.t = \text{temperature and } \{r_1.t\} = \{\text{skyCover}\}]\!] \\
\textbf{(R3)} & [\![r_2.t = \text{temperature and } r_j.t \neq \text{temperature } \forall j < 2]\!] \\
\textbf{(R4)} & [\![r_2.t = \text{temperature and } r_2.v[\text{time}] = \texttt{5pm-6am}]\!] \\
 & [\![r_2.t = \text{temperature and } r_2.v[\text{min}] = \texttt{low}]\!] \\
 & [\![r_2.t = \text{temperature and } r_2.v[\text{mean}] = \texttt{low}]\!] \\
 & [\![r_2.t = \text{temperature and } r_2.v[\text{max}] = \texttt{medium}]\!]
\end{array}
$$

Angeli et al 2010: Generation Example

$$F_2 = \{\text{time, min}\}$$

(F1) $[\![F_2 = \{\text{time}, \text{min}\}]\!]$
(F2) $[\![F_2 = \{\text{time}, \text{min}\} \text{ and } r_2.v[\text{time}] = \texttt{5pm-6am}]\!]$
(F2) $[\![F_2 = \{\text{time}, \text{min}\} \text{ and } r_2.v[\text{min}] = \texttt{low}]\!]$

Angeli et al 2010: Generation Example

$\mathcal{T}_2 =<$ *with a low around [min] .* $>$

$$
\begin{array}{ll}
\textbf{(W1)} & [\![\textsc{Base}(\mathcal{T}_2) = \langle \textit{with a low around} \text{ [min]}\rangle]\!] \\
& [\![\textsc{Coarse}(\mathcal{T}_2) = \langle \textit{with a} \text{ [time] } \textit{around} \text{ [min]}\rangle]\!] \\
\textbf{(W2)} & [\![\textsc{Base}(\mathcal{T}_2) = \langle \textit{with a low around} \text{ [min]}\rangle \text{ and } r_2.v[\text{time}] = \texttt{5pm-6am}]\!] \\
& [\![\textsc{Coarse}(\mathcal{T}_2) = \langle \textit{with a} \text{ [time] } \textit{around} \text{ [min]}\rangle \text{ and } r_2.v[\text{time}] = \texttt{5pm-6am}]\!] \\
& [\![\textsc{Base}(\mathcal{T}_2) = \langle \textit{with a low around} \text{ [min]}\rangle \text{ and } r_2.v[\text{min}] = \texttt{low}]\!] \\
& [\![\textsc{Coarse}(\mathcal{T}_2) = \langle \textit{with a} \text{ [time] } \textit{around} \text{ [min]}\rangle \text{ and } r_2.v[\text{min}] = \texttt{low}]\!] \\
\textbf{(W3)} & \log p_{\text{LM}}(\textit{with} \mid \textit{cloudy ,})
\end{array}
$$

Angeli et al 2010: Evaluation

- WeatherGov results:

System	F_1	BLEU*	English Fluency	Semantic Correctness
BASELINE	78.7	24.8	4.28 ± 0.78	4.15 ± 1.14
OURSYSTEM	79.9	28.8	4.34 ± 0.69	4.17 ± 1.21
WASPER-GEN	72.0	28.7	4.43 ± 0.76	4.27 ± 1.15
HUMAN	—	—	4.43 ± 0.69	4.30 ± 1.07

- Evaluates using BLEU.
  - Improves on WeatherGov, state-of-the-art on the rest.
  - Uses perfect input to evaluate BLEU (perfect Content Selection).
- F1 for Content Selection.

Angeli et al 2010: Evaluation Example

HUMAN

Records:	pass_1		
Fields:	arg1=purple10		arg2=purple9
Text:	*purple10*	*passes back to*	*purple9*

BASELINE

Records:	pass_1		
Fields:	arg1=purple10		arg2=purple9
Text:	*purple10*	*kicks to*	*purple9*

OURSYSTEM

Records:	pass_1		
Fields:	arg1=purple10		arg2=purple9
Text:	*purple10*	*passes to*	*purple9*

WASPER-GEN Text: *purple10 passes to purple9*

NLG as Parsing

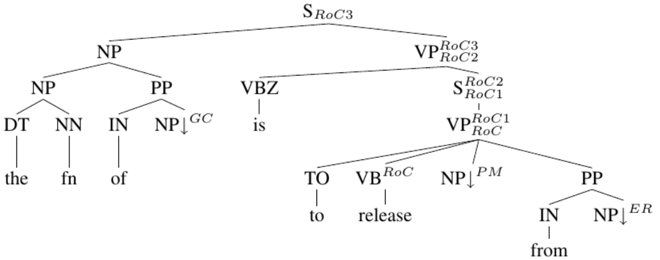
- Gyawali, B., & Gardent, C. *Surface Realization from Knowledge-Bases*. ACL-2014.
- KBGen domain.
- Tree Adjoining grammars parser plus semantic equations.
- Distill a semantic generation grammar and edit it by hand / generalize it automatically.
- Need to understand the formalism to be able to use it.

Gyawali & Gardent 2014: KBGen

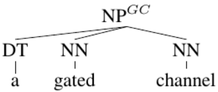
The function of a gated channel is to release particles from the endoplasmic reticulum

```
:TRIPLES (
(|Release-Of-Calcium646| |object| |Particle-In-Motion64582|)
(|Release-Of-Calcium646| |base| |Endoplasmic-Reticulum64603|)
(|Gated-Channel64605|  |has-function||Release-Of-Calcium646|)
(|Release-Of-Calcium646|  |agent| |Gated-Channel64605|))
:INSTANCE-TYPES
(|Particle-In-Motion64582| |instance-of| |Particle-In-Motion|)
(|Endoplasmic-Reticulum64603| |instance-of| |Endoplasmic-Reticulum|)
(|Gated-Channel64605| |instance-of| |Gated-Channel|)
 |Release-Of-Calcium646| |instance-of| |Release-Of-Calcium|))
:ROOT-TYPES (
(|Release-Of-Calcium646| |instance-of| |Event|)
(|Particle-In-Motion64582| |instance-of| |Entity|)
(|Endoplasmic-Reticulum64603| |instance-of| |Entity|)
(|Gated-Channel64605| |instance-of| |Entity|)))
```

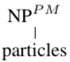
Gyawali & Gardent 2014: Extracted Grammar



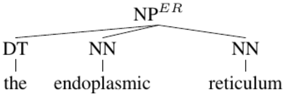
instance-of(RoC,Release-of-Calcium)
object(RoC,PM)
base(RoC,ER)
has-function(GC,RoC)
agent(RoC,GC)



instance-of(GC,Gated-Channel)



instance-of(PM,Particle-In-Motion)



instance-of(ER,Endoplasmic-Reticulum)

Gyawali & Gardent 2014: Results

System	All	Covered	Coverage	# Trees
IMS	0.12	0.12	100%	
UDEL	0.32	0.32	100%	
Base	0.04	0.39	30.5%	371
ManExp	0.28	0.34	83 %	412
AutExp	0.29	0.29	100%	477

- IMS: Statistical system using probabilistic grammar induced from data.
- UDEL: symbolic system from University of Delaware.
- Base: LTAG from corpora.
- MaxExp: Base + manual expansion (83% coverage).
- AutExp: Base + automatic expansion (close to UDEL performance).

Outline

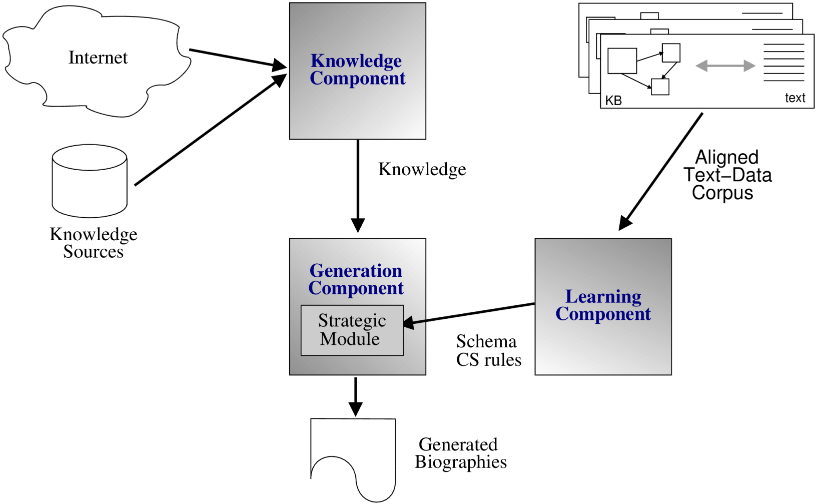
Intelligence Analysis

- ProGenIE was developed as part of the AQuAInt program.
  - Run by ARDA.
  - Multiple sites.
  - Consortia of universities and companies.
- Open research.
- Domain is biographies generation.
  - Using data-driven techniques.
  - Generate immediate up-to-date biographical profiles.
    - Different, Learned Content Plans.
    - Different, Possible Users.
- Research focus was on finding paradigmatic information to be included in the biography.
  - Average case, not the outliers.

ProGenIE

1. Information Extraction.
2. Content Selection rules.
3. Document Structuring schemata.
4. Lexical lookup.
5. Pronominalization.
6. Surface Realization using FUF generator.

ProGenIE: Architecture



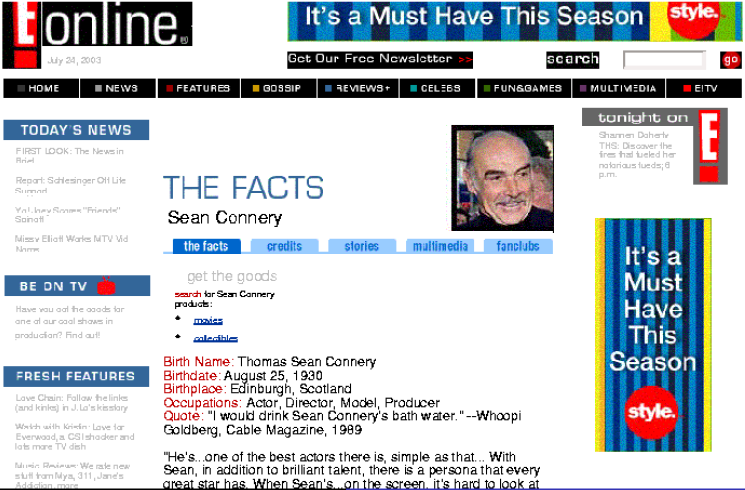
Information Extraction

```
Phase: Event
Input: Token Lookup Location Organization Date JobTitle Person
Options: control=appelt

Rule: e_r1
( ({Token.category == "DT"})?
  (({Organization})+)?
  ( ({Token.category == "JJ"})*
    ({Token.string == "chemical"}{Token.string == "weapons"})
    ({Token.string == "actions"}|{Token.string == "action"}|
     {Token.string == "attacks"}|{Token.string == "attack"})
  )
  {Token.string == "against"}
  ({Token.category == "DT"})?
  (({Location})+|
   ({Organization})+|
   (({Lookup.majorType == citizenship})+
    ({Token.string == "forces"}|{Token.string == "force"}|{Token.string == "army"}|
    {Token.string == "troops"}|{Token.string == "soldiers"}|{Token.string == "nationals"
    {Token.string == "interests"}|{Token.string == "citizens"}|{Token.string == "embassy
    {Token.string == "embassies"}|{Token.string == "consulate"}|{Token.string == "consul
    {Token.string == "diplomat"}|{Token.string == "diplomats"})))|
   (({Location})+ {Token.string == "'s"}
    {Token.string == "secret"}{Token.string == "service"}{Token.string == "headquarters"
   ):target
  ( {Token.string == "in"} (({Location})+) :place )?
  ( ({Token.string == "in"}|{Token.string == "during"}) (({Date})+) :date )?
):event --> { /* ... */ }
```

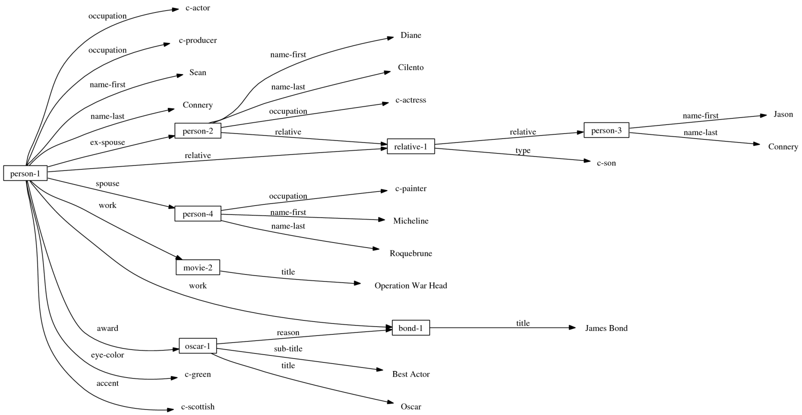
Approximating Knowledge Graphs

- Crawled ~1,000 factsheets from E! Entertainment TV



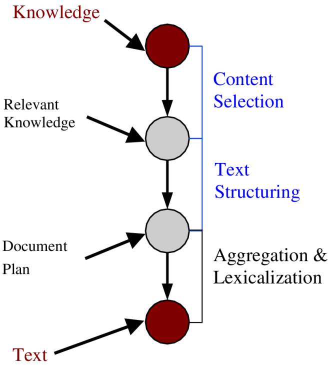
Approximating Knowledge Graphs

- Custom Perl scripts to extract knowledge graphs:



ProGenIE: Research

- Indirect Supervised Learning.
  - Now it will be called “weakly supervised”.
- Unsupervised migration of labels.
- Supervised training of subsystems.



ProGenIE: Content Selection

- Input: Set of Attribute Value Pairs.

⟨name first⟩	John	⟨name last⟩	Doe
⟨weight⟩	150Kg	⟨height⟩	160cm
⟨occupation⟩	c-writer	⟨occupation⟩	c-producer
⟨award title⟩	BAFTA	⟨award year⟩	1999
⟨relative type⟩	c-grandson	⟨rel. firstN⟩	Dashiel
⟨rel. lastN⟩	Doe	⟨rel. birthD⟩	1990

- Output: Selected Attribute-Value Pairs.

⟨**name first**⟩	John	⟨**name last**⟩	Doe
⟨**occupation**⟩	c-writer	⟨**occupation**⟩	c-producer

- Example Verbalization:

John Doe is a writer, producer, . . .

ProGenIE: Learning Problem

- Input: Set of Attribute Value Pairs.

⟨`name first`⟩	John	⟨`name last`⟩	Doe
⟨`weight`⟩	150Kg	⟨`height`⟩	160cm

← · · · →

John Doe, American writer, born in Maryland in 1967, famous for his strong prose and . . .

- Output: Content Selection rules.

`TRUE()` Always select.
Example: for node ∈ **name→last**, **select node**.

`IN(list of values)` Select if the value is in the list.
Example: for node ∈ **education→place→country**, if node_value ∈ {*"Scotland"*, *"England"*}, then **select node**.

`TRAVERSE(path,recursive-rule)` Select if the node at the end of the path matches the recursive-rule.
Example: for node ∈ **relative→relative→name→first**,

ProGenIE: Solution

- Unsupervised learning to label the KB triples.
- Stochastic search to learn the Content Selection rules.
- Learning rules for different target biography sources, ranging from one sentence to one full page.
  - The rules for one source are different from another source.
    - Capture editorial behaviour.

ProGenIE: Unsupervised Learning

- Given:
  - $(KB_1, Bio_1), (KB_2, Bio_2), (KB_3, Bio_3), (KB_4, Bio_4)$
- Cluster Knowledge Bases By Value:
  - $\{KB_1, KB_2\}$ contain $(\langle \texttt{birth} \rightarrow \texttt{place} \rightarrow \texttt{state} \rangle, `MD')$
  - $\{KB_3, KB_4\}$ contain $(\langle \texttt{birth} \rightarrow \texttt{place} \rightarrow \texttt{state} \rangle, `NY')$
- Compare Language Models Of Clusters:
  - Compare the models of $\{Bio_1, Bio_2\}$ against $\{Bio_3, Bio_4\}$.
  - If the models differ, select $\langle \texttt{birth} \rightarrow \texttt{place} \rightarrow \texttt{state} \rangle$.
- $Bio_1 \Rightarrow$ "…born in Maryland…"
- $Bio_2 \Rightarrow$ "…from Maryland…"
- $Bio_3 \Rightarrow$ "…native of New York…"
- $Bio_4 \Rightarrow$ "…born in New York…"

ProGenIE: Results

Experiment	biography.com				imdb.com			
	Selected	Prec.	Rec.	F*	Selected	Prec.	Rec.	F*
random	162	0.29	0.48	0.36	369	0.25	0.50	0.33
select-all	1129	0.26	1.00	0.41	1584	0.23	1.00	0.37
true/false rules	550	0.41	0.94	0.58	891	0.36	0.88	0.51
only exact match	359	0.64	0.61	0.62	432	0.48	0.65	0.55
combined	292	0.57	0.81	0.67	432	0.49	0.68	0.57
test set	293	-	-	-	369	-	-	-

Summary

- Mimicking the progress in NLU, NLG got its share of successes using statistical methods.
  - Slow start, due to the difficulty acquiring training data.
  - Problems also evaluating NLG output.
- Humans expect perfection when it comes to text.
  - Difficult to find niches where poor output will be accepted.
    - Machine Translation shines there.
- Some techniques are well established:
  - Language models.
  - Generate-and-rank.
  - Log-linear approaches.

- Outlook
  - Compared to NLU, NLG had a shorter run with statistical methods and moved directly to Deep Learning approaches.
  - We will see these techniques shortly.

For Further Reading

- Krahmer, E., Theune, M.
  Empirical Methods in Natural Language Generation.
  Springer, Berlin 2010.
- Bangalore, S., Stent, A.
  Natural Language Generation in Interactive Systems.
  Cambridge University Press, 2014.