Towards a noisy channel approach to QA

Towards a noisy channel approach to QA

- General problems of QA
  - efficient systems are very complex (e.g., a typical multi-stream/source architecture)
  - difficult to optimize/understand sub-parts because they are often connected in a non-trivial way
  - need transparent models that capture different aspects of QA (question classification, passage reranking, answer extraction) in a single framework

Towards a noisy channel approach to QA

- General problems of QA
  - efficient systems are very complex (e.g., a typical multi-stream/source architecture)
  - difficult to optimize/understand sub-parts because they are often connected in a non-trivial way
  - need transparent models that capture different aspects of QA (question classification, passage reranking, answer extraction) in a single framework

$\Rightarrow$ **noisy channel**

Towards a noisy channel approach to QA

- General problems of QA
  - efficient systems are very complex (e.g., a typical multi-stream/source architecture)
  - difficult to optimize/understand sub-parts because they are often connected in a non-trivial way
  - need transparent models that capture different aspects of QA (question classification, passage reranking, answer extraction) in a single framework

  $\Rightarrow$ **noisy channel**

- Noisy channel idea is similar to language modeling in IR:
  - LM in IR: $\mathrm{argmax}_{\boldsymbol{d}} \boldsymbol{P}(\boldsymbol{d}|\boldsymbol{q}) =_{(\text{uniform priors})} \mathrm{argmax}_{\boldsymbol{d}} \boldsymbol{P}(\boldsymbol{q}|\boldsymbol{M_d})$
  - $\boldsymbol{P}(\boldsymbol{q}|\boldsymbol{M_d})$: probability that query $\boldsymbol{q}$ is written by the author of $\boldsymbol{d}$

Towards a noisy channel approach to QA

- General problems of QA
  - efficient systems are very complex (e.g., a typical multi-stream/source architecture)
  - difficult to optimize/understand sub-parts because they are often connected in a non-trivial way
  - need transparent models that capture different aspects of QA (question classification, passage reranking, answer extraction) in a single framework

  $\Rightarrow$ **noisy channel**

- Noisy channel idea is similar to language modeling in IR:
  - LM in IR: $\mathrm{argmax}_{\boldsymbol{d}} \boldsymbol{P}(\boldsymbol{d}|\boldsymbol{q}) =_{(\text{uniform priors})} \mathrm{argmax}_{\boldsymbol{d}} \boldsymbol{P}(\boldsymbol{q}|\boldsymbol{M_d})$
  - $\boldsymbol{P}(\boldsymbol{q}|\boldsymbol{M_d})$: probability that query $\boldsymbol{q}$ is written by the author of $\boldsymbol{d}$
  - QA via Noisy Channel model: $\mathrm{argmax}_{\boldsymbol{s}} \boldsymbol{P}(\boldsymbol{q}|\boldsymbol{s})$
  - $\boldsymbol{P}(\boldsymbol{q}|\boldsymbol{s})$: probability that question $\boldsymbol{q}$ can be asked about (or is answered by) sentence $\boldsymbol{s}$

General noisy channel idea

General noisy channel idea

- source emits true sequences (T)

General noisy channel idea

- source emits true sequences (T)
- distorted sequences are observed (O) at the destination

General noisy channel idea

- source emits true sequences ($\boldsymbol{T}$)
- distorted sequences are observed ($\boldsymbol{O}$) at the destination
- given $\boldsymbol{O}$, what was the original $\boldsymbol{T}$? $\mathrm{argmax}_{\boldsymbol{T}}\boldsymbol{P}(\boldsymbol{T}|\boldsymbol{O})$?

General noisy channel idea

- source emits true sequences (T)
- distorted sequences are observed (O) at the destination
- given O, what was the original T? $\mathrm{argmax}_T P(T|O)$?
- Apply Bayes rule: $P(T|O) = \frac{P(O|T) \cdot P(T)}{P(O)}$

General noisy channel idea

- source emits true sequences (T)
- distorted sequences are observed (O) at the destination
- given O, what was the original T? $\mathrm{argmax}_T P(T|O)$?
- Apply Bayes rule: $P(T|O) = \frac{P(O|T) \cdot P(T)}{P(O)}$
  - $P(O)$ is constant

General noisy channel idea

- source emits true sequences (T)
- distorted sequences are observed (O) at the destination
- given O, what was the original T? $\mathrm{argmax}_T P(T|O)$?
- Apply Bayes rule: $P(T|O) = \frac{P(O|T) \cdot P(T)}{P(O)}$
  - $P(O)$ is constant
  - $P(T)$ assumed constant for simplicity (although in principle we can make the model reflect that e.g. "a lazy dog" is more likely in our language than "lazy a dog"; i.e., we can introduce a "true language" model)

General noisy channel idea

- source emits true sequences (T)
- distorted sequences are observed (O) at the destination
- given O, what was the original T? $\text{argmax}_T P(T|O)$?
- Apply Bayes rule: $P(T|O) = \frac{P(O|T) \cdot P(T)}{P(O)}$
  - $P(O)$ is constant
  - $P(T)$ assumed constant for simplicity (although in principle we can make the model reflect that e.g. "a lazy dog" is more likely in our language than "lazy a dog"; i.e., we can introduce a "true language" model)
  - $P(O|T)$ is the noise (distortion) model

General noisy channel idea

- source emits true sequences (T)
- distorted sequences are observed (O) at the destination
- given O, what was the original T? $\text{argmax}_T P(T|O)$?
- Apply Bayes rule: $P(T|O) = \frac{P(O|T) \cdot P(T)}{P(O)}$
  - $P(O)$ is constant
  - $P(T)$ assumed constant for simplicity (although in principle we can make the model reflect that e.g. "a lazy dog" is more likely in our language than "lazy a dog"; i.e., we can introduce a "true language" model)
  - $P(O|T)$ is the noise (distortion) model
- **therefore:** $\text{argmax}_T P(T|O) \sim \text{argmax}_T P(O|T)$

General noisy channel idea

- source emits true sequences (T)
- distorted sequences are observed (O) at the destination
- given O, what was the original T? $\mathrm{argmax}_T P(T|O)$?
- Apply Bayes rule: $P(T|O) = \frac{P(O|T) \cdot P(T)}{P(O)}$
  - $P(O)$ is constant
  - $P(T)$ assumed constant for simplicity (although in principle we can make the model reflect that e.g. "a lazy dog" is more likely in our language than "lazy a dog"; i.e., we can introduce a "true language" model)
  - $P(O|T)$ is the noise (distortion) model
- therefore: $\mathrm{argmax}_T P(T|O) \sim \mathrm{argmax}_T P(O|T)$
- **$P(O|T)$ estimated from training data**

Applications of the noisy channel idea

Applications of the noisy channel idea

- Speech recognition: text "distorted" into audio signal
  - Audio signal A is observed. What was the text T?

Applications of the noisy channel idea

- Speech recognition: text "distorted" into audio signal
  - Audio signal A is observed. What was the text T?
  - estimate phonetic model $P(A|T)$, e.g., using Markov model: $\prod_i P(A_i|T_i, T_{i-1}, T_{i-2})$

Applications of the noisy channel idea

- Speech recognition: text "distorted" into audio signal
  - Audio signal A is observed. What was the text T?
  - estimate phonetic model $P(A|T)$, e.g., using Markov model: $\prod_i P(A_i|T_i, T_{i-1}, T_{i-2})$
  - the unit (A_i) is problem: sound? syllable?

Applications of the noisy channel idea

- Speech recognition: text "distorted" into audio signal
  - Audio signal A is observed. What was the text T?
  - estimate phonetic model $P(A|T)$, e.g., using Markov model: $\prod_i P(A_i|T_i, T_{i-1}, T_{i-2})$
  - the unit (A_i) is problem: sound? syllable?

- OCR: text "distorted" into printed/written char sequence
  - Printed chars C observed. What was the original text T?

Applications of the noisy channel idea

- Speech recognition: text ”distorted” into audio signal
  - Audio signal A is observed. What was the text T?
  - estimate phonetic model $P(A|T)$, e.g., using Markov model: $\prod_i P(A_i|T_i, T_{i-1}, T_{i-2})$
  - the unit (A_i) is problem: sound? syllable?

- OCR: text “distorted” into printed/written char sequence
  - Printed chars C observed. What was the original text T?
  - $P(C|T)$ estimated from training data as $\prod_i P(C_i|T_i)$
    - $P(C_i|T_i)$ defines “how each character is written”

Applications of the noisy channel idea

- Speech recognition: text "distorted" into audio signal
  - Audio signal A is observed. What was the text T?
  - estimate phonetic model $P(A|T)$, e.g., using Markov model: $\prod_i P(A_i|T_i, T_{i-1}, T_{i-2})$
  - the unit (A_i) is problem: sound? syllable?
- OCR: text "distorted" into printed/written char sequence
  - Printed chars C observed. What was the original text T?
  - $P(C|T)$ estimated from training data as $\prod_i P(C_i|T_i)$
    - $P(C_i|T_i)$ defines "how each character is written"
  - Key is finding the right unit (C_i): e.g., bitmap of what size? starting where?

Applications of the noisy channel idea

- Speech recognition: text ”distorted” into audio signal
  - Audio signal A is observed. What was the text T?
  - estimate phonetic model $P(A|T)$, e.g., using Markov model: $\prod_i P(A_i|T_i, T_{i-1}, T_{i-2})$
  - the unit (A_i) is problem: sound? syllable?
- OCR: text “distorted” into printed/written char sequence
  - Printed chars C observed. What was the original text T?
  - $P(C|T)$ estimated from training data as $\prod_i P(C_i|T_i)$
    - $P(C_i|T_i)$ defines “how each character is written”
  - Key is finding the right unit (C_i): e.g., bitmap of what size? starting where?
  - By setting non-trivial model for $P(T)$ we can use language-specific OCR and thus perform error correction

Another application: Machine Translation

Another application: Machine Translation

- Machine translation (MT):
  - Translation into another language as distortion
  - O is original sentence (observed), T is the translation we are looking for
  - Many models to estimate $P(O|T)$

Another application: Machine Translation

- Machine translation (MT):
  - Translation into another language as distortion
  - O is original sentence (observed), T is the translation we are looking for
  - Many models to estimate $P(O|T)$
- Example: classic MT model IBM-4 (Brown, 1993)

Another application: Machine Translation

- Machine translation (MT):
  - Translation into another language as distortion
  - O is original sentence (observed), T is the translation we are looking for
  - Many models to estimate $P(O|T)$
- Example: classic MT model IBM-4 (Brown, 1993)
  - IBM-4: every word in (O, NULL) generates 0, 1 or more words of T

Another application: Machine Translation

- Machine translation (MT):
  - Translation into another language as distortion
  - O is original sentence (observed), T is the translation we are looking for
  - Many models to estimate $P(O|T)$
- Example: classic MT model IBM-4 (Brown, 1993)
  - IBM-4: every word in (O, NULL) generates 0, 1 or more words of T
  - Ex. "Hoe heet je?" $\longrightarrow$ "What is your name?"

Another application: Machine Translation

- Machine translation (MT):
  - Translation into another language as distortion
  - O is original sentence (observed), T is the translation we are looking for
  - Many models to estimate $P(O|T)$

- Example: classic MT model IBM-4 (Brown, 1993)
  - IBM-4: every word in (O, NULL) generates 0, 1 or more words of T
  - Ex. "Hoe heet je?" $\longrightarrow$ "What is your name?"
    - possible alignment: Hoe$\longrightarrow$What, heet$\longrightarrow$is, heet$\longrightarrow$name, je$\longrightarrow$your, ?$\longrightarrow$?
    - $P(O|T, alignment) = P(Hoe|What) \cdot P(heet|is) \cdot \ldots$

Another application: Machine Translation

- Machine translation (MT):
  - Translation into another language as distortion
  - O is original sentence (observed), T is the translation we are looking for
  - Many models to estimate $P(O|T)$
- Example: classic MT model IBM-4 (Brown, 1993)
  - IBM-4: every word in (O, NULL) generates 0, 1 or more words of T
  - Ex. "Hoe heet je?" $\longrightarrow$ "What is your name?"
    - possible alignment: Hoe⟶What, heet⟶is, heet⟶name, je⟶your, ?⟶?
    - $P(O|T, alignment) = P(Hoe|What) \cdot P(heet|is) \cdot \ldots$
  - $P(O|T) = \mathrm{argmax}_a P(O|T, a)$ — we find the best alignment using **Viterbi search**

Application to QA

Application to QA

- Applying noisy channel model to QA
  - Questions are generated from sentences, i.e., questions are “distorted” declarative sentences
  - $P(Q|S) = ?$

Application to QA

- Applying noisy channel model to QA
  - Questions are generated from sentences, i.e., questions are "distorted" declarative sentences
  - $P(Q|S) = ?$
  - **Example**
    - Q: *When did Elvis Presley die?*
    - Answer sentence S: *Presley died of heart disease at Graceland in 1977, and . . .*

Application to QA

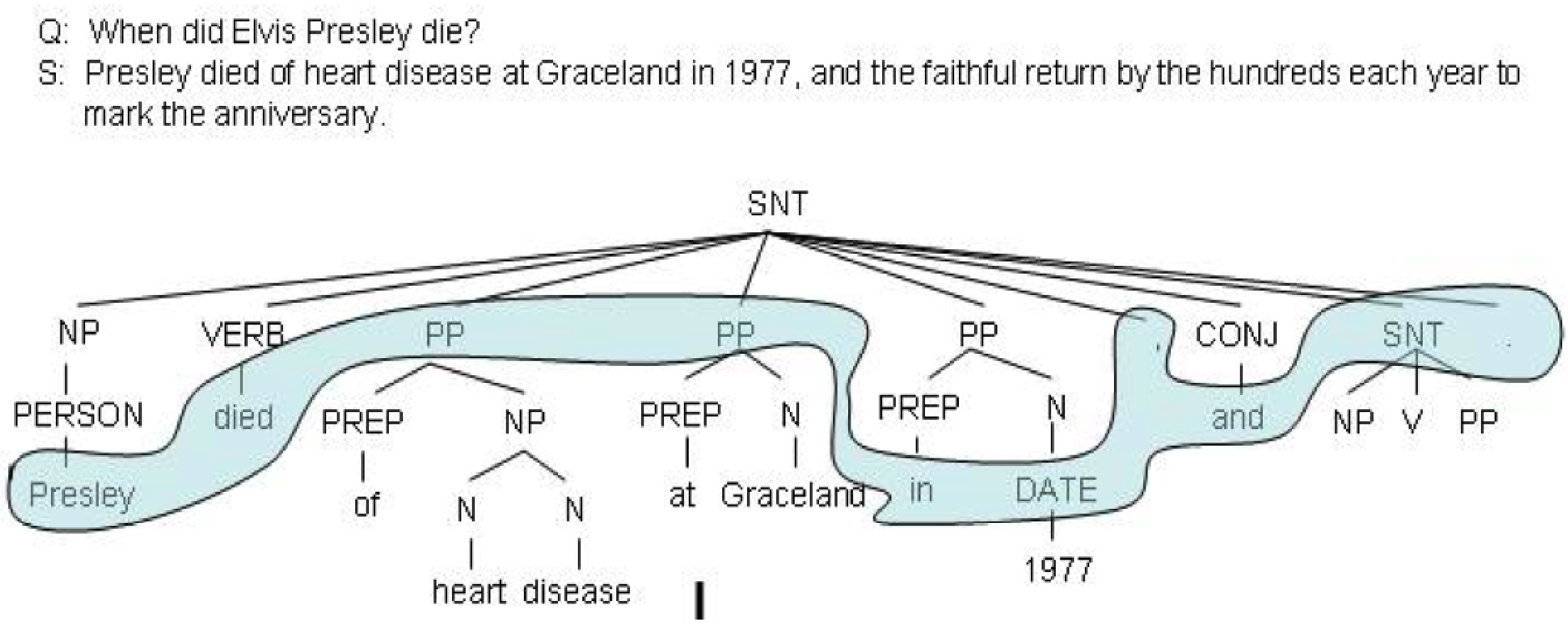
- Applying noisy channel model to QA
  - Questions are generated from sentences, i.e., questions are "distorted" declarative sentences
  - $P(Q|S) = ?$
  - **Example**
    - Q: *When did Elvis Presley die?*
    - Answer sentence S: *Presley died of heart disease at Graceland in 1977, and . . .*
  - What kind of "distortions"?
    - *Presley* $\longrightarrow$ *Elvis Presley*
    - *died* $\longrightarrow$ *did . . . die*
    - *in 1977* $\longrightarrow$ *When*
    - [the rest of the sentence goes to NULL]

Application to QA

- Applying noisy channel model to QA
  - Questions are generated from sentences, i.e., questions are “distorted” declarative sentences
  - $P(Q|S) =?$
  - **Example**
    - Q: *When did Elvis Presley die?*
    - Answer sentence S: *Presley died of heart disease at Graceland in 1977, and . . .*
  - What kind of “distortions”?
    - *Presley* $\longrightarrow$ *Elvis Presley*
    - *died* $\longrightarrow$ *did . . . die*
    - *in 1977* $\longrightarrow$ *When*
    - [the rest of the sentence goes to NULL]
  - Answer sentences are typically longer than corresponding questions: whole phrases are mapped to NULL

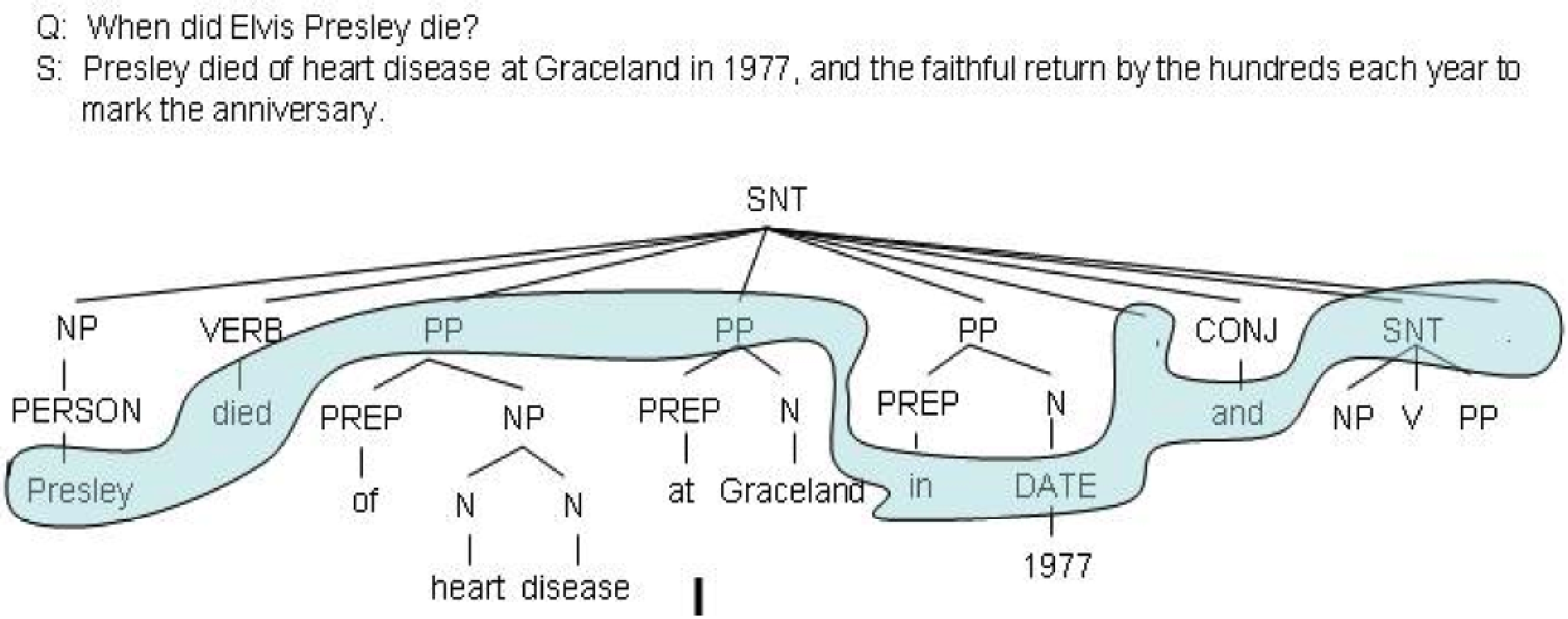
Question generation (sentence distortion) model

Question generation (sentence distortion) model



- Make a syntactic parse of S: identify (nested) phrases ($\rightsquigarrow$ Erik's talk)

Question generation (sentence distortion) model

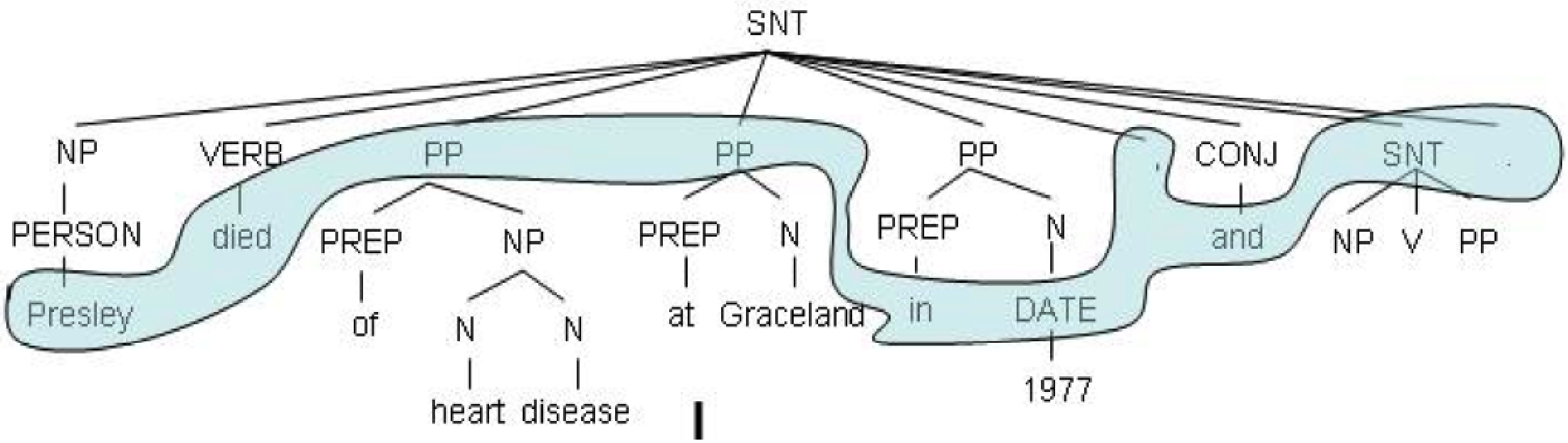


- Make a syntactic parse of S: identify (nested) phrases ($\rightsquigarrow$ Erik's talk)
- Identify semantic entities (PERSON, DATE, etc.) - Named Entity Tagging

Question generation (2)

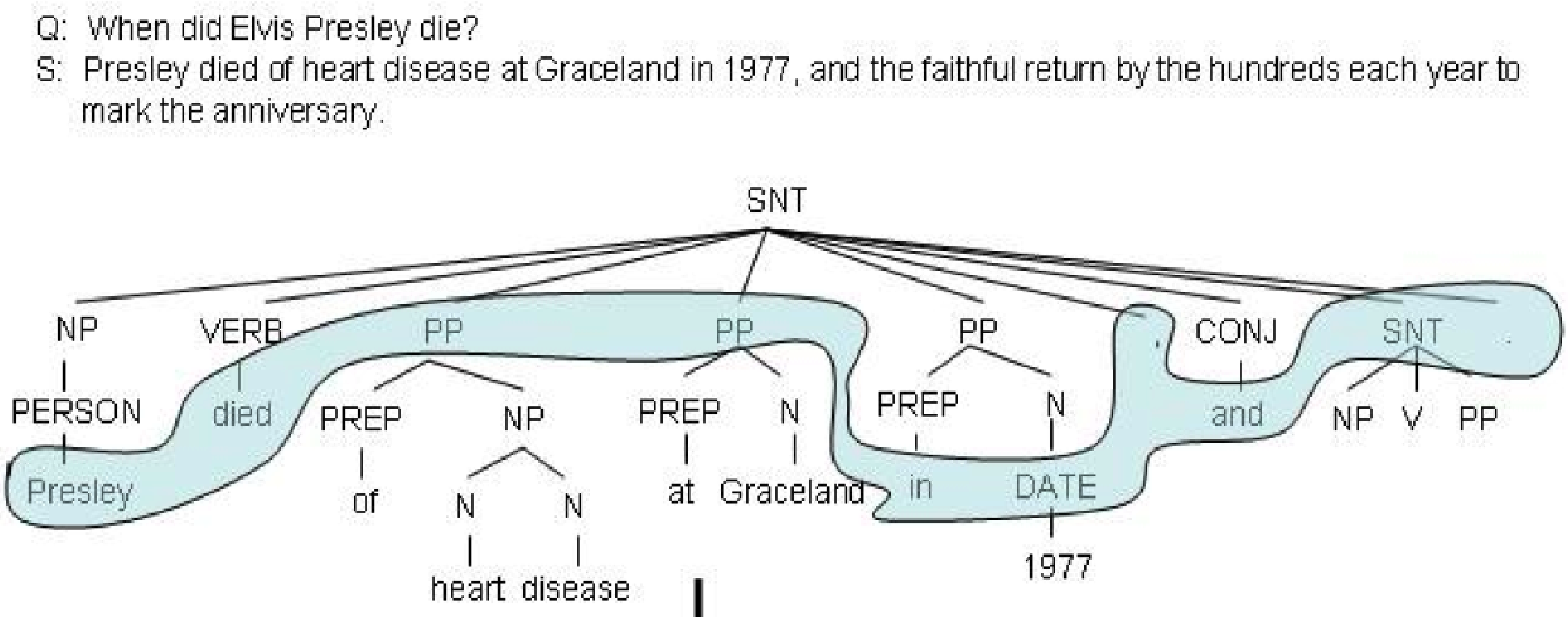
Question generation (2)

Q: When did Elvis Presley die?
S: Presley died of heart disease at Graceland in 1977, and the faithful return by the hundreds each year to mark the anniversary.



- Make horizontal cut through parse tree (all cuts equally likely)

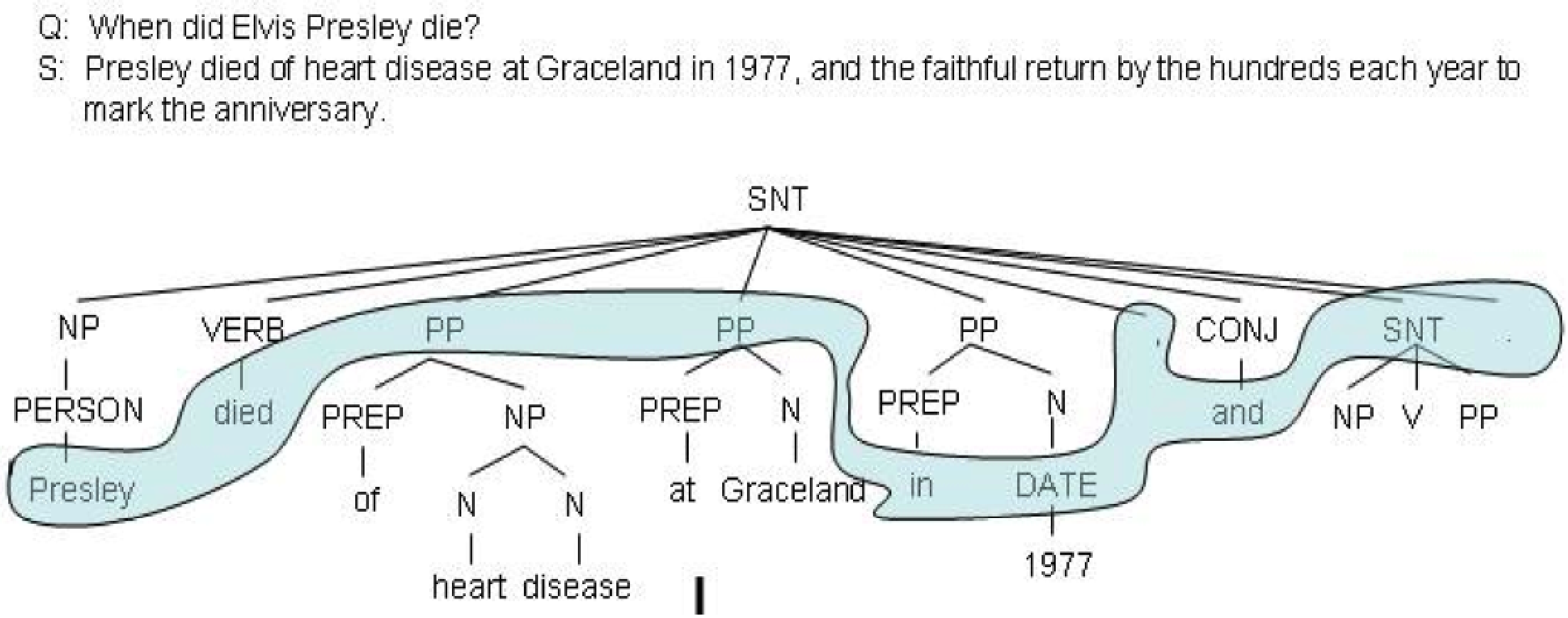
Question generation (2)



- Make horizontal cut through parse tree (all cuts equally likely)
- Read off the labels in the cut
  - *Presley : died : PP : PP : in : DATE : , : and : SNT : .*

Question generation (3)

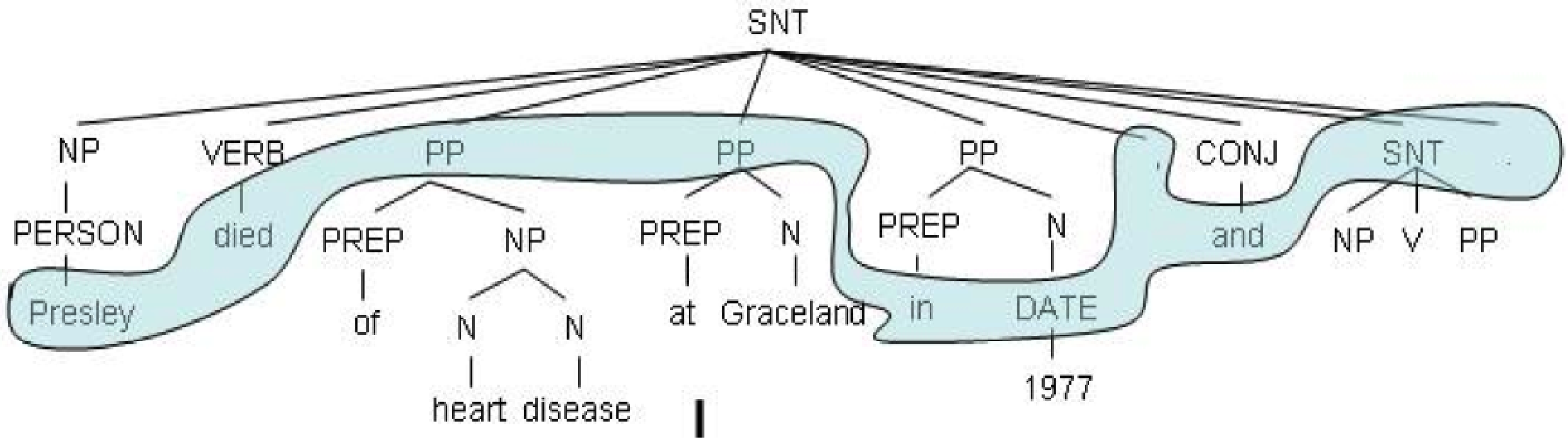
Question generation (3)



- Mark one label as potential answer (all labels equally likely)
  - *Presley : died : PP : PP : in : A_DATE : , : and : SNT : .*

Question generation (3)

Q: When did Elvis Presley die?
S: Presley died of heart disease at Graceland in 1977, and the faithful return by the hundreds each year to mark the anniversary.



- Mark one label as potential answer (all labels equally likely)
  - *Presley : died : PP : PP : in : A_DATE : , : and : SNT : .*
- Run resulting sequence through noisy channel (e.g., IBM model 4)

Question generation (4)

Question generation (4)

- ***S* :** *Presley : died : PP : PP : in : A_DATE : , : and : SNT : .*

Question generation (4)

- S **:** *Presley : died : PP : PP : in : A_DATE : , : and : SNT : .*
- Q **:** *When : did : Elvis : Presley : die : ?*

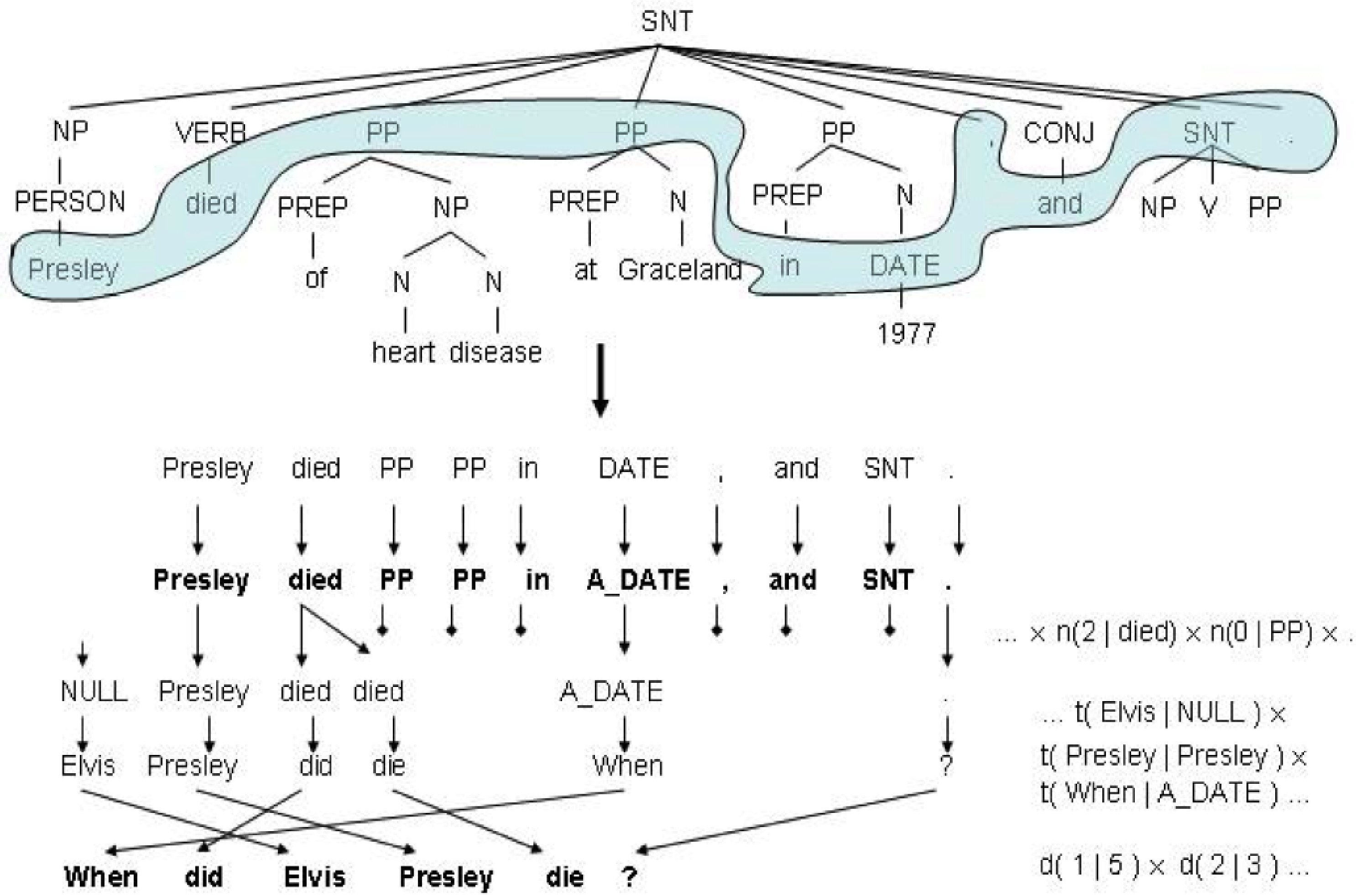
Question generation (4)

- S **:** *Presley : died : PP : PP : in : A_DATE : , : and : SNT : .*
- Q **:** *When : did : Elvis : Presley : die : ?*
- Some labels are skipped, some added, e.g.
  - Presley ⟶ Elvis Presley
  - died ⟶ did . . . die
  - PP PP ⟶ NULL
  - in ⟶ NULL
  - A_DATE ⟶ When
  - , ⟶ NULL
  - SNT ⟶ NULL

Question generation in a glance

Q: When did Elvis Presley die?

S: Presley died of heart disease at Graceland in 1977, and the faithful return by the hundreds each year to mark the anniversary.



Putting it to work

Putting it to work

- Estimating the model
  - Training corpus corpus of pairs (Q, S) with answers marked
    - E.g., TREC QA data, questions/answers from quizzes
  - Parse S'es
  - Make "good" cuts in syntactic trees by aligning words of Q's and S'es
  - Use existing implementation of MT models to learn translation probabilities

Putting it to work

- Estimating the model
  - Training corpus corpus of pairs (Q, S) with answers marked
    - E.g., TREC QA data, questions/answers from quizzes
  - Parse S'es
  - Make "good" cuts in syntactic trees by aligning words of Q's and S'es
  - Use existing implementation of MT models to learn translation probabilities

- And finally, **answering questions**
  - Incoming question Q and candidate sentences S_1, . . . , S_n obtained using IR engine
  - $\mathrm{argmax}_i P(Q|S_i) = ?$
  - The method does work with just 1,000-2,000 questions (20,000–50,000 Q/A pairs)

Wrapping up

Wrapping up

- Flexibility of the model
  - Synonyms and "related" words can be added via translation probabilities
    - $\boldsymbol{P(purchase|buy) = high}$
    - available from resources like WordNet
  - Questions can be paraphrased using rule-based methods, increasing the number of training Q/A pairs
  - Reliable structured resources (CIA World Factbook, Biography.com,. . .) can be used to generate Q/A pairs automatically

Wrapping up

- Flexibility of the model
  - Synonyms and "related" words can be added via translation probabilities
    - $P(purchase|buy) = high$
    - available from resources like WordNet
  - Questions can be paraphrased using rule-based methods, increasing the number of training Q/A pairs
  - Reliable structured resources (CIA World Factbook, Biography.com,. . .) can be used to generate Q/A pairs automatically

- Noisy channel QA
  - Approach reuses well-known ideas and software
  - Straightforward integration of different resources
  - No need for question classification
  - . . .