Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

NLG, Wrap up

Scott Farrar
CLMA, University of Washington
farrar@u.washington.edu

March 10, 2010

Today's lecture

1. Statistical NLG
2. Surface realizer
   - Linearization
3. SimpleNLG
   - Lexicon
4. Design ideas

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

NLG research

Methods

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

NLG research

Methods

- **canned text**: predefined utterances are returned based on the string value

NLG research

Methods

- **canned text**: predefined utterances are returned based on the string value
- **template-based**: hard-coded templates are filled in w missing constituents

NLG research

Methods

- **canned text**: predefined utterances are returned based on the string value
- **template-based**: hard-coded templates are filled in w missing constituents
- **statistical**: corpus is used to construct a language model

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

NLG research

Methods

- **canned text**: predefined utterances are returned based on the string value
- **template-based**: hard-coded templates are filled in w missing constituents
- **statistical**: corpus is used to construct a language model
- **hybrid approach**: use templates, but select best candidate based on corpus.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Goals of Statistical NLG

In a fully **statistical NLG system**, a text is produced that is as close to a given language model as possible, with a couple of goals in mind:

Goals of Statistical NLG

In a fully **statistical NLG system**, a text is produced that is as close to a given language model as possible, with a couple of goals in mind:

- **fidelity**: that the output is as faithfully representative to human text as possible

Goals of Statistical NLG

In a fully **statistical NLG system**, a text is produced that is as close to a given language model as possible, with a couple of goals in mind:

- **fidelity**: that the output is as faithfully representative to human text as possible
- **fluency**: that the language generated is more easily processed and naturalistic

Scott Farrar CLMA, University of Washington farrar@u.washington.ed

Which is the text produced from Bigram?

, armed in their stings , Make boot upon the summer ' s velvet buds , Which pillage they with merry march bring home to the tent - royal of their emperor ; Who , busied in his majesty , surveys The singing masons building roofs of gold , The civil citizens kneading up the honey , The poor mechanic porters crowding in Their heavy burdens at his narrow gate , The sad - ey 'd justice , with his surly hum , Delivering o ' er to executors pale The lazy yawning drone .

retreat . By my hand into the clouds , as cold as any military man in divers functions , Setting endeavour in continual motion ; To - morrow is Saint Crispian . He trots the air , a subject for a sovereign ' s ambition ! He was thinking of civil wars when he speaks , The King hath heard that men of mould . Abate thy rage . Use lenity , sweet chuck . NYM . They know your worthiness . My liege , as you shall read that my Nell is dead i ' faith , my cousin Suffolk

Which is the text produced from Bigram?

, armed in their stings , Make boot upon the summer ' s velvet buds , Which pillage they with merry march bring home to the tent - royal of their emperor ; Who , busied in his majesty , surveys The singing masons building roofs of gold , The civil citizens kneading up the honey , The poor mechanic porters crowding in Their heavy burdens at his narrow gate , The sad - ey 'd justice , with his surly hum , Delivering o ' er to executors pale The lazy yawning drone .

retreat . By my hand into the clouds , as cold as any military man in divers functions , Setting endeavour in continual motion ; To - morrow is Saint Crispian . He trots the air , a subject for a sovereign ' s ambition ! He was thinking of civil wars when he speaks , The King hath heard that men of mould . Abate thy rage . Use lenity , sweet chuck . NYM . They know your worthiness . My liege , as you shall read that my Nell is dead i ' faith , my cousin Suffolk

Bigram

Which is the text produced from Trigram?

, armed in their stings , Make boot upon the summer ' s velvet buds , Which pillage they with merry march bring home to the tent - royal of their emperor ; Who , busied in his majesty , surveys The singing masons building roofs of gold , The civil citizens kneading up the honey , The poor mechanic porters crowding in Their heavy burdens at his narrow gate , The sad - ey 'd justice , with his surly hum , Delivering o ' er to executors pale The lazy yawning drone .

HENRY . We are in God 's peace ! I have an excellent armour ; but in loving me you should love the lovely bully . What mean have defeated the law ; Who when they were as cold as any ' s ambition ! He was thinking of civil wars when he was a merry message . KING HENRY . Thou doest thy office fairly . Turn head and stop pursuit ; for we hear Your greeting is from him , you men of mould . Abate thy rage , abate they manly rage ; Abate thy rage ,

Which is the text produced from Trigram?

, armed in their stings , Make boot upon the summer ’ s velvet buds , Which pillage they with merry march bring home to the tent - royal of their emperor ; Who , busied in his majesty , surveys The singing masons building roofs of gold , The civil citizens kneading up the honey , The poor mechanic porters crowding in Their heavy burdens at his narrow gate , The sad - ey ’d justice , with his surly hum , Delivering o ’ er to executors pale The lazy yawning drone .

HENRY . We are in God ’s peace ! I have an excellent armour ; but in loving me you should love the lovely bully . What mean have defeated the law ; Who when they were as cold as any ’ s ambition ! He was thinking of civil wars when he was a merry message . KING HENRY . Thou doest thy office fairly . Turn head and stop pursuit ; for we hear Your greeting is from him , you men of mould . Abate thy rage , abate they manly rage ; Abate thy rage ,

Trigram

Which is the text produced from Trigram?

> *, armed in their stings , Make boot upon the summer ' s velvet buds , Which pillage they with merry march bring home to the tent - royal of their emperor ; Who , busied in his majesty , surveys The singing masons building roofs of gold , The civil citizens kneading up the honey , The poor mechanic porters crowding in Their heavy burdens at his narrow gate , The sad - ey 'd justice , with his surly hum , Delivering o ' er to executors pale The lazy yawning drone .*

Shakespeare

> *HENRY . We are in God 's peace ! I have an excellent armour ; but in loving me you should love the lovely bully . What mean have defeated the law ; Who when they were as cold as any ' s ambition ! He was thinking of civil wars when he was a merry message . KING HENRY . Thou doest thy office fairly . Turn head and stop pursuit ; for we hear Your greeting is from him , you men of mould . Abate thy rage , abate they manly rage ; Abate thy rage ,*

Trigram

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Fluency goals

Fluency

Achieved according to **macroscopic properties**, or those properties of text that describe non-content issues:

- sentence length
- vocabulary diversity
- use of certain syntactic structures (relatives, lists)
- surface stylistics (commas, punc., capitalization)

All things being equal, text *A* and text *B* could be produced with mostly different macroscopic properties, yet they would both represent the same information.

Fluency goals

Fluency

Achieved according to **macroscopic properties**, or those properties of text that describe non-content issues:

- sentence length
- vocabulary diversity
- use of certain syntactic structures (relatives, lists)
- surface stylistics (commas, punc., capitalization)

All things being equal, text *A* and text *B* could be produced with mostly different macroscopic properties, yet they would both represent the same information.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Fluency goals

Fluency

Achieved according to **macroscopic properties**, or those properties of text that describe non-content issues:

- sentence length
- vocabulary diversity
- use of certain syntactic structures (relatives, lists)
- surface stylistics (commas, punc., capitalization)

All things being equal, text *A* and text *B* could be produced with mostly different macroscopic properties, yet they would both represent the same information.

Fluency goals

Fluency

Achieved according to **macroscopic properties**, or those properties of text that describe non-content issues:

- sentence length
- vocabulary diversity
- use of certain syntactic structures (relatives, lists)
- surface stylistics (commas, punc., capitalization)

All things being equal, text *A* and text *B* could be produced with mostly different macroscopic properties, yet they would both represent the same information.

Fluency goals

Fluency

Achieved according to **macroscopic properties**, or those properties of text that describe non-content issues:

- sentence length
- vocabulary diversity
- use of certain syntactic structures (relatives, lists)
- surface stylistics (commas, punc., capitalization)

All things being equal, text *A* and text *B* could be produced with mostly different macroscopic properties, yet they would both represent the same information.

Fluency goals

Fluency

Achieved according to **macroscopic properties**, or those properties of text that describe non-content issues:

- sentence length
- vocabulary diversity
- use of certain syntactic structures (relatives, lists)
- surface stylistics (commas, punc., capitalization)

All things being equal, text *A* and text *B* could be produced with mostly different macroscopic properties, yet they would both represent the same information.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Comparing macroscopic properties

Example

Man, that was a sweet deal you made. What was that guy thinking?

Example

Dude, you really scored with that deal. He was a real sucker.

Are these equal?

Comparing macroscopic properties

Example

Man, that was a sweet deal you made. What was that guy thinking?

Example

Dude, you really scored with that deal. He was a real sucker.

Are these equal?

Comparing macroscopic properties

Example

Man, that was a sweet deal you made. What was that guy thinking?

Example

Dude, you really scored with that deal. He was a real sucker.

Are these equal?

Pure statistical NLG

But, a pure fully statistical NLG engine would be of minimal use. It would produce isolated utterances that sounded fine, but might be odd. Why?

- Domain, subject matter is highly specific.
- Context is completely lost.
- Turn doesn't match previous (in dialogue).

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Pure statistical NLG

But, a pure fully statistical NLG engine would be of minimal use. It would produce isolated utterances that sounded fine, but might be odd. Why?

- Domain, subject matter is highly specific.
- Context is completely lost.
- Turn doesn't match previous (in dialogue).

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Pure statistical NLG

But, a pure fully statistical NLG engine would be of minimal use. It would produce isolated utterances that sounded fine, but might be odd. Why?

- Domain, subject matter is highly specific.
- Context is completely lost.
- Turn doesn't match previous (in dialogue).

Pure statistical NLG

But, a pure fully statistical NLG engine would be of minimal use. It would produce isolated utterances that sounded fine, but might be odd. Why?

- Domain, subject matter is highly specific.
- Context is completely lost.
- Turn doesn't match previous (in dialogue).

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Hybrid NLG

Hybrid techniques, on the other hand, provide the best of both worlds:

- template-based NLG to ensure relevance (fidelity)
- corpus-based NLG to produce natural sounding utterances (fluency). For example, content planning can still be accomplished using symbolic techniques. But condition upon the domain/genre:
    - choose lexical items (*heart* vs. *ticker*)
    - chose referring exp. syntax: *the large black dog*, *that big dog*, *the black one*
    - match dialogue act with tense

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Hybrid NLG

Hybrid techniques, on the other hand, provide the best of both worlds:

- template-based NLG to ensure relevance (fidelity)
- corpus-based NLG to produce natural sounding utterances (fluency). For example, content planning can still be accomplished using symbolic techniques. But condition upon the domain/genre:
    - choose lexical items (*heart* vs. *ticker*)
    - chose referring exp. syntax:
      the large black dog, *that big dog*, *the black one*
    - match dialogue act with tense

Scott Farrar
CLMA, University of Washington far-
rar@u.washington.ed

Hybrid NLG

Hybrid techniques, on the other hand, provide the best of both worlds:

- template-based NLG to ensure relevance (fidelity)
- corpus-based NLG to produce natural sounding utterances (fluency). For example, content planning can still be accomplished using symbolic techniques. But condition upon the domain/genre:
  - choose lexical items (*heart* vs. *ticker*)
  - chose referring exp. syntax:
    the large black dog, that big dog, the black one
  - match dialogue act with tense

Hybrid NLG

Hybrid techniques, on the other hand, provide the best of both worlds:

- template-based NLG to ensure relevance (fidelity)
- corpus-based NLG to produce natural sounding utterances (fluency). For example, content planning can still be accomplished using symbolic techniques. But condition upon the domain/genre:
  - choose lexical items (*heart* vs. *ticker*)
  - chose referring exp. syntax: *the large black dog, that big dog, the black one*
  - match dialogue act with tense

Hybrid NLG

Hybrid techniques, on the other hand, provide the best of both worlds:

- template-based NLG to ensure relevance (fidelity)
- corpus-based NLG to produce natural sounding utterances (fluency). For example, content planning can still be accomplished using symbolic techniques. But condition upon the domain/genre:
    - choose lexical items (*heart* vs. *ticker*)
    - chose referring exp. syntax:
      the large black dog, *that big dog*, *the black one*
    - match dialogue act with tense

Hybrid NLG

Hybrid techniques, on the other hand, provide the best of both worlds:

- template-based NLG to ensure relevance (fidelity)
- corpus-based NLG to produce natural sounding utterances (fluency). For example, content planning can still be accomplished using symbolic techniques. But condition upon the domain/genre:
    - choose lexical items (*heart* vs. *ticker*)
    - chose referring exp. syntax:
      the large black dog, *that big dog*, *the black one*
    - match dialogue act with tense

Templates in Hybrid NLG

Given a text, extract potential templates:

- I'd like to leave **Houston at 5pm**.
- Can you recommend a good **wine** ?
- I wanna order **a sandwich** .

Now transform utterances into templates and fill with domain-specific items:

- I'd like to leave <CITY> at <TIME> .
- Can you recommend a good <PRONOUN> ?
- I wanna order <FOOD> .

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Templates in Hybrid NLG

Given a text, extract potential templates:

- I'd like to leave **Houston at 5pm**.
- Can you recommend a good **wine** ?
- I wanna order **a sandwich** .

Now transform utterances into templates and fill with domain-specific items:

- I'd like to leave <CITY> at <TIME> .
- Can you recommend a good <PRONOUN> ?
- I wanna order <FOOD> .

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Templates in Hybrid NLG

Given a text, extract potential templates:

- I'd like to leave **Houston at 5pm**.
- Can you recommend a good **wine** ?
- I wanna order **a sandwich** .

Now transform utterances into templates and fill with domain-specific items:

- I'd like to leave <CITY> at <TIME> .
- Can you recommend a good <PRONOUN> ?
- I wanna order <FOOD> .

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Templates in Hybrid NLG

Given a text, extract potential templates:

- I'd like to leave **Houston at 5pm**.
- Can you recommend a good **wine** ?
- I wanna order **a sandwich** .

Now transform utterances into templates and fill with domain-specific items:

- I'd like to leave <CITY> at <TIME> .
- Can you recommend a good <PRONOUN> ?
- I wanna order <FOOD> .

Scott Farrar
CLMA, University of Washington far-
rar@u.washington.ed

Templates in Hybrid NLG

Given a text, extract potential templates:

- I'd like to leave **Houston at 5pm**.
- Can you recommend a good **wine** ?
- I wanna order **a sandwich** .

Now transform utterances into templates and fill with domain-specific items:

- I'd like to leave <CITY> at <TIME> .
- Can you recommend a good <PRONOUN> ?
- I wanna order <FOOD> .

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Templates in Hybrid NLG

Given a text, extract potential templates:

- I'd like to leave **Houston at 5pm**.
- Can you recommend a good **wine** ?
- I wanna order **a sandwich** .

Now transform utterances into templates and fill with domain-specific items:

- I'd like to leave <CITY> at <TIME> .
- Can you recommend a good <PRONOUN> ?
- I wanna order <FOOD> .

Scott Farrar
CLMA, University of Washington far-
rar@u.washington.ed

Templates in Hybrid NLG

Given a text, extract potential templates:

- I'd like to leave **Houston at 5pm**.
- Can you recommend a good **wine** ?
- I wanna order **a sandwich** .

Now transform utterances into templates and fill with domain-specific items:

- I'd like to leave <CITY> at <TIME> .
- Can you recommend a good <PRONOUN> ?
- I wanna order <FOOD> .

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Templates in Hybrid NLG

Given a text, extract potential templates:

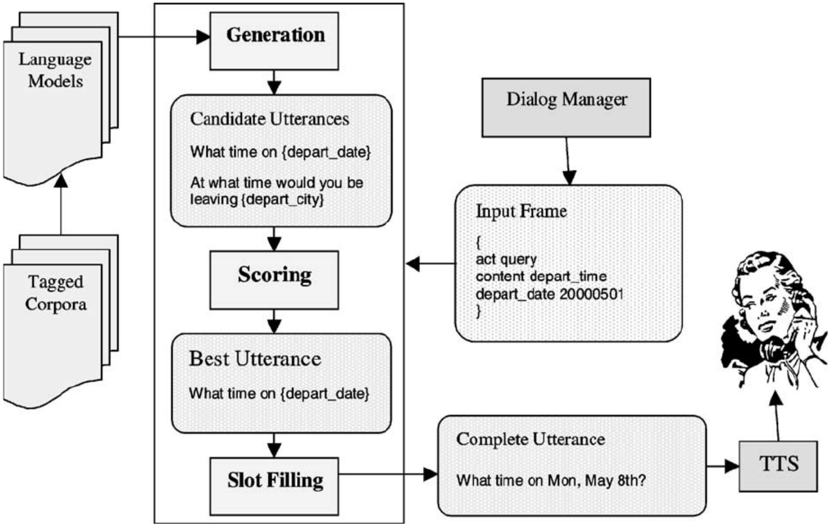
- I'd like to leave **Houston at 5pm**.
- Can you recommend a good **wine** ?
- I wanna order **a sandwich** .

Now transform utterances into templates and fill with domain-specific items:

- I'd like to leave <CITY> at <TIME> .
- Can you recommend a good <PRONOUN> ?
- I wanna order <FOOD> .

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Components in statistical NLG



Today's lecture

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Surface realizer: Purpose

To generate natural language strings from a fully specified input (deterministic); the inverse of certain kinds of parsing processes.

- determines the surface form of the text;
- adds inflectional endings of words;
- orders constituents;
- misc. markup (e.g., lists, paragraphs, punctuation)

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Surface realizer: Inputs/Outputs

Input:

- phrase specifications
- Or for an entire text, a text specification

Output:

- linearized sentences, texts

Incremental NLG

A surface realizer adds more and more grammatical detail:

1. lexical items
2. morphosyntactic info
3. surface form with inflection
4. punctuation, capitalization (intonation if spoken)

Example

1. request itinerary
2. 2.SG POSS request INDEF.itinerary
3. you can request an itinerary
4. You can request an itinerary.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Incremental NLG

A surface realizer adds more and more grammatical detail:

1. lexical items
2. morphosyntactic info
3. surface form with inflection
4. punctuation, capitalization (intonation if spoken)

Example

1. request itinerary
2. 2.SG POSS request INDEF.itinerary
3. you can request an itinerary
4. You can request an itinerary.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Incremental NLG

A surface realizer adds more and more grammatical detail:

1. lexical items
2. morphosyntactic info
3. surface form with inflection
4. punctuation, capitalization (intonation if spoken)

Example

1. request itinerary
2. 2.SG POSS request INDEF.itinerary
3. you can request an itinerary
4. You can request an itinerary.

Scott Farrar CLMA, University of Washington farrar@u.washington.ed

Incremental NLG

A surface realizer adds more and more grammatical detail:

1. lexical items
2. morphosyntactic info
3. surface form with inflection
4. punctuation, capitalization (intonation if spoken)

Example

1. request itinerary
2. 2.SG POSS request INDEF.itinerary
3. you can request an itinerary
4. You can request an itinerary.

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Scott Farrar CLMA, University of Washington farrar@u.washington.ed

Incremental NLG

A surface realizer adds more and more grammatical detail:

1. lexical items
2. morphosyntactic info
3. surface form with inflection
4. punctuation, capitalization (intonation if spoken)

Example

1. request itinerary
2. 2.SG POSS request INDEF.itinerary
3. you can request an itinerary
4. You can request an itinerary.

Surface realizer

main functions:

- **linguistic realization**: uses rules of grammar (about morphology and syntax) to convert abstract representations of sentences into actual text.
- **structure realization**: converts abstract structures such as paragraphs and sentences into mark-up (punctuated text, HTML, etc.)

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Linearization

The microplanner identifies and specifies the order of constituents, but does not put the constituents in the final order.

It's left up to the surface realization component to carry out the instructions encoded in the phrase specification:

- English: adjectivals before nouns, e.g., *giant tortoise*
- Spanish: adjectivals after nouns, e.g., *tortuga gigante*

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Today's lecture

1. Statistical NLG
2. Surface realizer
   - Linearization
3. SimpleNLG
   - Lexicon
4. Design ideas

Library contents

Key coomponents of SimpleNLG

- `simplenlg.features`: various morphosyntactic and discourse features
- `simplenlg.framework`: key NLG elements (documents, phrases, words)
- `simplenlg.lexicon`: the lexicon class
- `simplenlg.realiser.english`: the actual realiser

Library contents

Key coomponents of SimpleNLG

- `simplenlg.features`: various morphosyntactic and discourse features
- `simplenlg.framework`: key NLG elements (documents, phrases, words)
- `simplenlg.lexicon`: the lexicon class
- `simplenlg.realiser.english`: the actual realiser

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Library contents

Key coomponents of SimpleNLG

- `simplenlg.features`: various morphosyntactic and discourse features
- `simplenlg.framework`: key NLG elements (documents, phrases, words)
- `simplenlg.lexicon`: the lexicon class
- `simplenlg.realiser.english`: the actual realiser

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Library contents

Key coomponents of SimpleNLG

- `simplenlg.features`: various morphosyntactic and discourse features
- `simplenlg.framework`: key NLG elements (documents, phrases, words)
- `simplenlg.lexicon`: the lexicon class
- `simplenlg.realiser.english`: the actual realiser

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Library contents

Key coomponents of SimpleNLG

- `simplenlg.features`: various morphosyntactic and discourse features
- `simplenlg.framework`: key NLG elements (documents, phrases, words)
- `simplenlg.lexicon`: the lexicon class
- `simplenlg.realiser.english`: the actual realiser

Scott Farrar
CLMA, University of Washington far-rar@u.washington.ed

Process

Microplanning

```
PhraseElement myNP = phraseFactory.createNounPhrase(w);

DocumentElement sentence2 = documentFactory.createSentence();

sentence2.addComponent(myNP);
```

Realization

```
Realiser realiser = new Realiser();

realiser.setLexicon(lexicon);

NLGElement mydoc = realiser.realise(mydoc);

System.out.println(mydoc.getRealisation());
```

Scott Farrar CLMA, University of Washington farrar@u.washington.ed

Features and values available in SimpleNLG

Tense	Fut, Past, Pres
Person	First, Second, Third
Gender	Feminine, Masculine, Neuter
NumberAgr	Both, Plural, Singular
Pattern	Regular, Irregular, Regular_Double, ...
Interrogative	How, Where, Why, etc.
ClauseStatus	Matrix, Subordinate
DiscourseFuction	Cue_Phrase, Post_modifier, Complement, etc.

SimpleNLG lexicons

Available lexicons

- DefaultLexicon
- NIHLexicon

```
2010 Release:

432822 records
551454 baseForms
758153 forms
                  Item    BaseForms    Forms
                  ----    ---------   ------

noun            350050       430625   614737
adj              61999        93135    95089
verb             11001        14274    57412
adv               9416        13044    13108
prep               155          170      170
pron                87           88       88
conj                65           69       69
det                 38           38       38
modal                7            7       25
aux                  3            3       30
```

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Lexical entries

```
 WordElement: base=sell, category=VERB, {realisation=null,
      category=VERB, features={isDitransitive=true,
       presentParticiple=selling, present3s=sells,
       intransitive=true, transitive=true, pastParticiple=sold,
       past=sold}}

WordElement: base=Franklin, category=NOUN, {realisation=null,
   category=NOUN, features={proper=true, nonCount=false}}

WordElement: base=big, category=ADJECTIVE, {realisation=null,
    category=ADJECTIVE, features={isClassifyingAdj=false,
    comparative=bigger, predicative=true, superlative=biggest,
    isColourAdjective=false, isQualitativeAdjective=true}}
```

Today's lecture

Scott Farrar
CLMA, University of Washington farrar@u.washington.ed

Possible class hierarchies

You'll have 3 separate class hierarchies (with as much structure as you wish):

1. Messages, e.g., BirthMessage, DeathMessage
2. KB entities / things in the domain, e.g., Person, Location, etc.
3. SimpleNLG entities (phrases, whole docs, etc)

Methods

Create methods in the various message classes to output instances of `NLGElement`.