

SimpleNLG bilingue

L'ajout du français à un réalisateur de texte anglais

Pierre-Luc Vaudry

Qu'est-ce que SimpleNLG?

Qu'est-ce que SimpleNLG?

- Bibliothèque Java
 - 60 classes réparties dans 10 packages
 - 22 classes de tests JUnit
- Développée par Albert Gatt et Ehud Reiter de l'Université d'Aberdeen
- Réalisation de texte en anglais
- Utile dans le contexte de la génération de texte
- SimpleNLG bilingue est une adaptation de la version originale de SimpleNLG 4.2

Génération de texte

- Intelligence artificielle et linguistique informatique
- Entrée : données sous forme non linguistique
- Connaissance de la langue
- Connaissance du domaine
- Sortie : texte en langue naturelle
- Produit : rapports, messages d'aide, pages web, etc.

Génération de texte

- Exemples d'utilisation de la génération de texte :
 - Rapports météorologiques
 - Bulletins de prévisions météorologiques
 - Descriptions de modèles orientés objet
 - Pages web présentant des connaissances taxonomiques
 - Lettres personnalisées d'encouragement à l'arrêt du tabagisme
 - Résumés par abstraction d'articles de presse

Rôle de SimpleNLG

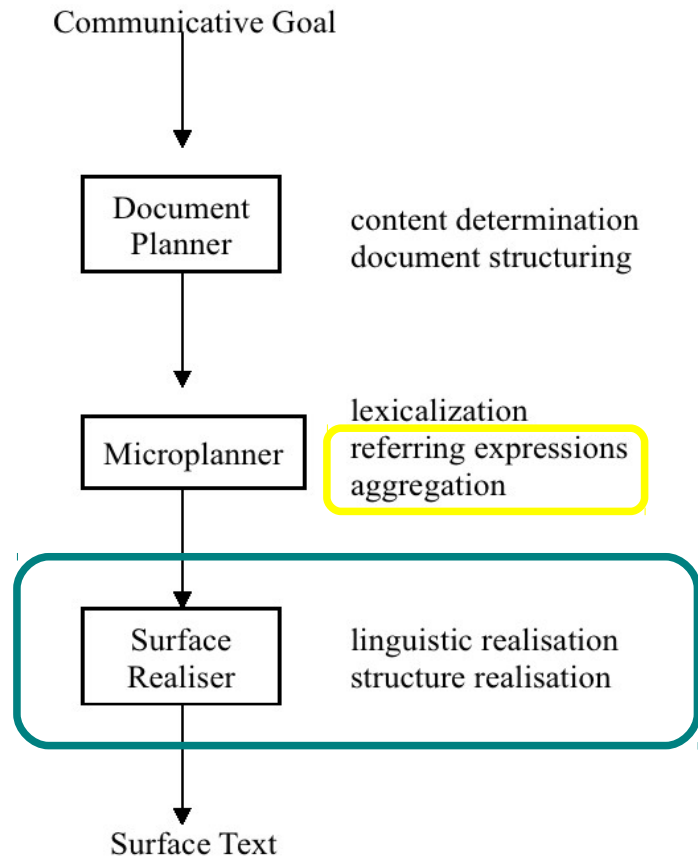


Figure 2: A typical NLG system architecture. From Reiter and Dale p.60.

- Effectue la réalisation de surface à partir d'une représentation abstraite d'un texte
 - Grammaire
 - Lexique
- Utile aussi pour
 - Agrégation
 - Anaphore (pronom personnel)

Rôle de SimpleNLG

- Entrée : représentation abstraite
 - Mot, syntagme, proposition, coordination
 - Phrase, paragraphe, liste, section, document
 - “Texte en boîte” (“*canned text*”)
 - Sous forme de :
 - arbre d'objets Java, de classes définies par SimpleNLG
 - Version 4.3 : document XML (non encore inclus dans SimpleNLG bilingue)
- Sortie : texte formaté

Capacités de SimpleNLG

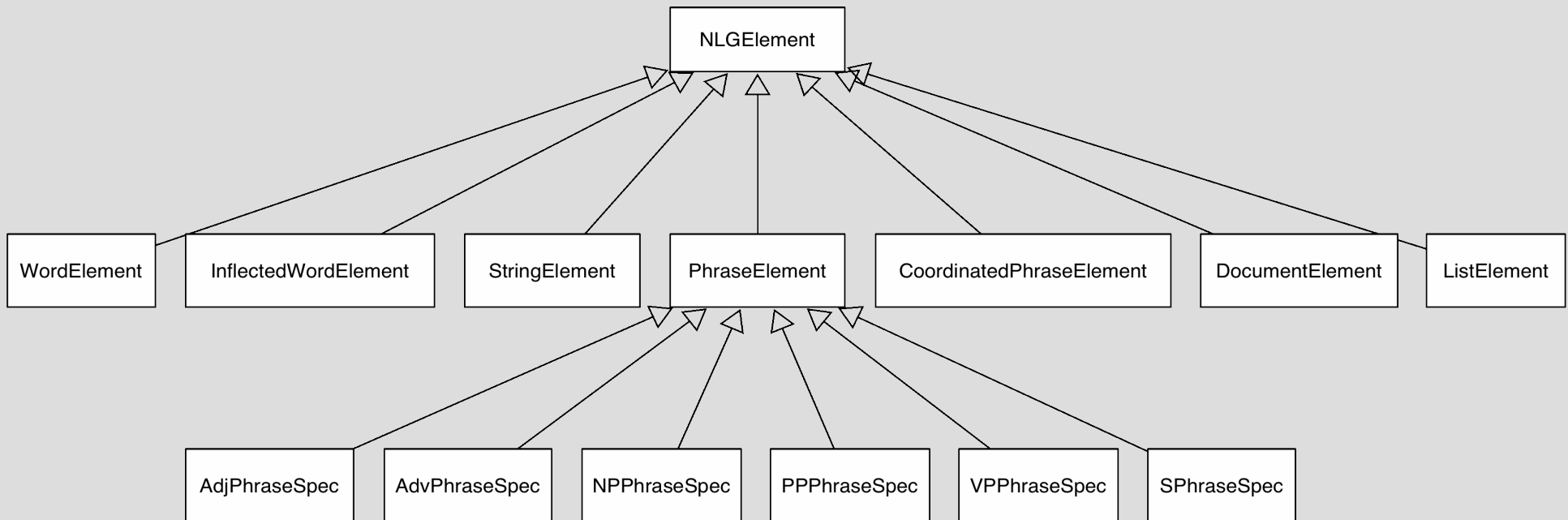
- Morphologie
 - Flexion : nom, verbe, pronom, déterminant
 - Dérivation : adjectif et adverbe
- Syntaxe
 - Ordre des mots
 - Accord
 - Auxiliaires : temps, voix, aspect
 - SN remplacé par un pronom personnel
 - Agrégation
- Ponctuation, espacement et majuscule

Fonctionnement de SimpleNLG

Fonctionnement de SimpleNLG

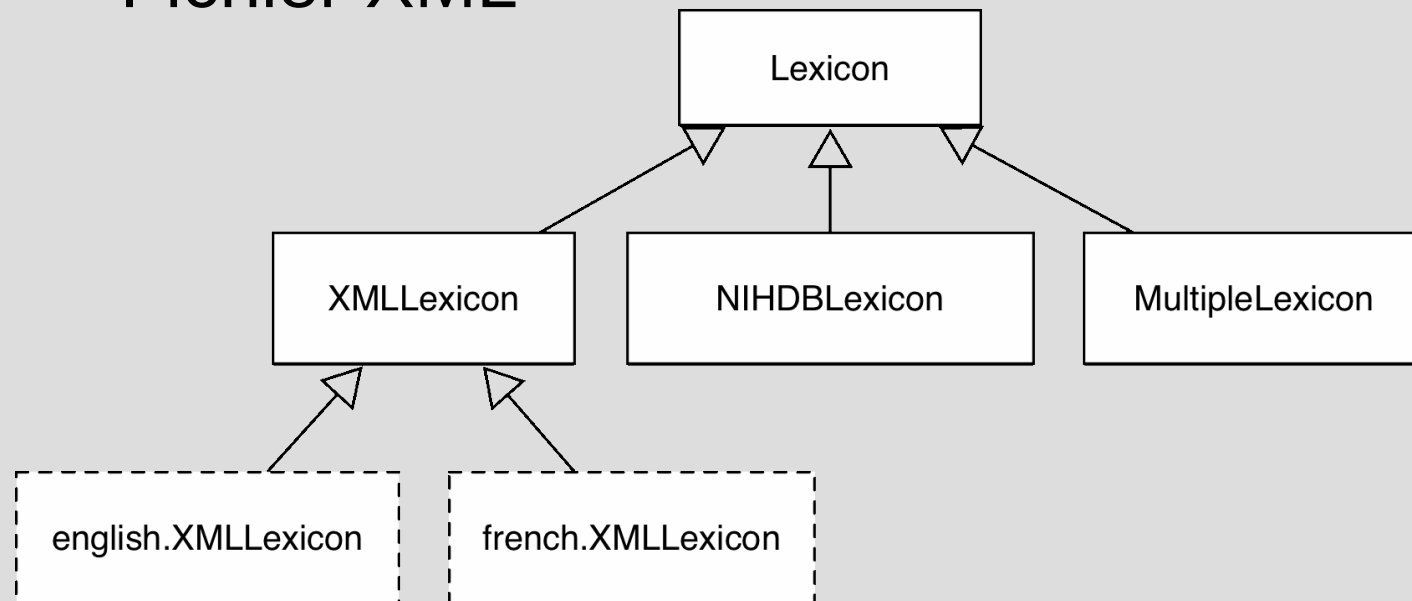
- Classes dérivées de NLGElement
 - Pour construire l'arbre à réaliser par SimpleNLG
- WordElement
 - Unités lexicales
 - Uniques
 - Appartiennent au lexique (Lexicon)
- Classes dérivées de PhraseElement
 - Tous types de syntagmes
- StringElement : “texte en boîte”
- DocumentElement
 - Phrase, paragraphe, section, liste, document

NLGElement



Lexique

- Offre l'accès à un ensemble d'unités lexicales (WordElement)
- Données lexicales provenant de :
 - Banque de données (NIH Specialist lexicon)
 - Fichier XML



Modificateurs et compléments

- Différentes façons d'ajouter un élément non central à un syntagme
- Modificateurs
 - Prémodificateur
 - Postmodificateur
 - Placement par défaut des modificateurs
 - Modificateur de tête de proposition
(*frontmodifier*)
- Complément
 - Placé immédiatement après la tête du syntagme

Traits (*features*)

- Permettent d'ajuster les propriétés des unités lexicales et des syntagmes
- Exemples :
 - Mettre au pluriel un syntagme nominal
 - Mettre une proposition au passé, au passif ou à la négative
 - Signaler qu'un syntagme nominal doit être remplacé par un pronom personnel
 - Régler les propriétés d'une nouvelle unité lexicale créée en cours d'exécution

Réalisation

- Quatre étapes :
 - Syntaxe
 - Morphologie
 - Orthographe
 - Ponctuation, espacement et majuscule initiale
 - Formatage
 - Paragraphes, titres et sous-titres
- Module d'agrégation appelé séparément

Exemple d'utilisation

```
package simplenlg.examples;

import simplenlg.framework.*;
import simplenlg.lexicon.*;
import simplenlg.realiser.english.*;
import simplenlg.phrasespec.*;

public class GreetCrowd {

    public static void main(String[] args) {

        Lexicon lexicon = new XMLLexicon();
        NLGFactory factory = new NLGFactory(lexicon);
        Realiser realiser = new Realiser(lexicon);

        NPPhraseSpec theMan = factory.createNounPhrase("the", "man");
        NPPhraseSpec theCrowd = factory.createNounPhrase("the", "crowd");

        SPhraseSpec greeting = factory.createClause(theMan, "greet", theCrowd);

        String outString = realiser.realiseSentence(greeting);
        System.out.println(outString);
    }
}
```

Sortie:

The man greets the crowd.

Ajout du français

Objectifs

- Produire du français grammatical
- Le moins de changements possible
 - Interface programmeur
 - Fonctionnement interne
 - Ajout de sous-classes
 - Modification des classes existantes
 - Utilisation des tests JUnit existants
- Permettre la génération bilingue
- Faciliter l'ajout futur d'autres langues

Tâches

- Séparer le général du spécifique à l'anglais
 - Grammaire
 - Lexique
- Langue de chaque élément
- Règles grammaticales françaises
- Lexique français

Défis

- Syntaxe
 - Syntagme nominal
 - Placement par défaut de l'adjectif
 - Accord du déterminant et de l'adjectif
 - Remplacement par un pronom personnel
 - Syntagme verbal et proposition
 - Négation (avec ou sans auxiliaire de négation)
 - Auxiliaire verbal
 - Placement des pronoms compléments clitiques
 - Accord du participe passé
 - Ordre des compléments non clitiques
 - Placement des modificateurs
 - Coordination

Défis

- Morphologie
 - Flexion
 - Pluriel : nom, déterminant, adjectif
 - Féminin : adjectif
 - Temps verbaux : imparfait, futur, impératif
 - Formes du pronom personnel
 - Dérivation
 - Comparatif
 - Nom de genre opposé

Défis

- Morphophonologie
 - Ajout d'une étape de traitement supplémentaire
 - Prépositions : à, de
 - Élision
 - Formes graphiques de liaison
- Ponctuation
 - Coordination
 - Modificateurs

Exemple en français

```
package simplenlg.examples;

import simplenlg.framework.*;
import simplenlg.lexicon.Lexicon;
import simplenlg.lexicon.french.XMLLexicon;
import simplenlg.realiser.*;
import simplenlg.phrasespec.*;

public class SaluerFoule {

    public static void main(String[] args) {

        Lexicon lexicon = new XMLLexicon();
        NLGFactory factory = new NLGFactory(lexicon);
        Realiser realiser = new Realiser();

        NPPhraseSpec theMan = factory.createNounPhrase("le", "homme");
        NPPhraseSpec theCrowd = factory.createNounPhrase("le", "foule");

        SPhraseSpec greeting = factory.createClause(theMan, "saluer", theCrowd);

        String outString = realiser.realiseSentence(greeting);
        System.out.println(outString);
    }
}
```

Sortie:

L'homme salue la foule.

Exemple de génération bilingue

```
public static void main(String[] args) {  
  
    Lexicon englishLexicon = new simplenlg.lexicon.english.XMLLexicon();  
    NLGFactory englishFactory = new NLGFactory(englishLexicon);  
  
    Lexicon frenchLexicon = new simplenlg.lexicon.french.XMLLexicon();  
    NLGFactory frenchFactory = new NLGFactory(frenchLexicon);  
  
    Realiser realiser = new Realiser();  
  
    Map<DictioEntry,WordElement> englishSensesMapping =  
        createEnglishSensesMapping(englishLexicon);  
    Map<DictioEntry,WordElement> frenchSensesMapping =  
        createFrenchSensesMapping(frenchLexicon);  
  
    SPhraseSpec englishGreeting = buildGreetCrowd(englishFactory, englishSensesMapping);  
    SPhraseSpec frenchGreeting = buildGreetCrowd(frenchFactory, frenchSensesMapping);  
  
    DocumentElement paragraph = englishFactory.createParagraph();  
    paragraph.addComponent(englishGreeting);  
    paragraph.addComponent(frenchGreeting);  
    DocumentElement document = englishFactory.createDocument("Bilingual greeting\n");  
    document.addComponent(paragraph);  
  
    String outString = realiser.realise(document).getRealisation();  
    System.out.println(outString);  
}
```


Exemple de génération bilingue

```
public static Map<DictioEntry, WordElement> createEnglishSensesMapping(
    Lexicon englishLexicon) {
    Map<DictioEntry, WordElement> englishSensesMapping =
        new EnumMap<DictioEntry, WordElement>(DictioEntry.class);

    englishSensesMapping.put(DictioEntry.ADULT_MALE,
        englishLexicon.getWord("man", LexicalCategory.NOUN));
    englishSensesMapping.put(DictioEntry.BILINGUAL,
        englishLexicon.getWord("bilingual", LexicalCategory.ADJECTIVE));
    englishSensesMapping.put(DictioEntry.CROWD,
        englishLexicon.getWord("crowd", LexicalCategory.NOUN));
    englishSensesMapping.put(DictioEntry.DEFINITE_DETERMINER,
        englishLexicon.getWord("the", LexicalCategory.DETERMINER));
    englishSensesMapping.put(DictioEntry.GREETING_VERB,
        englishLexicon.getWord("greet", LexicalCategory.VERB));
    englishSensesMapping.put(DictioEntry.LANGUAGE_PREPOSITION,
        englishLexicon.getWord("in", LexicalCategory.PREPOSITION));
    englishSensesMapping.put(DictioEntry.OLD,
        englishLexicon.getWord("old", LexicalCategory.ADJECTIVE));
    englishSensesMapping.put(DictioEntry.SPANISH_LANGUAGE,
        englishLexicon.getWord("Spanish", LexicalCategory.NOUN));
    englishSensesMapping.put(DictioEntry.YET,
        englishLexicon.getWord("yet", LexicalCategory.ADVERB));

    return englishSensesMapping;
}
```

Exemple de génération bilingue

```
private static Map<DictioEntry, WordElement> createFrenchSensesMapping(
    Lexicon frenchLexicon) {
    Map<DictioEntry, WordElement> frenchSensesMapping =
        new EnumMap<DictioEntry, WordElement>(DictioEntry.class);

    frenchSensesMapping.put(DictioEntry.ADLT_MALE,
        frenchLexicon.getWord("homme", LexicalCategory.NOUN));
    frenchSensesMapping.put(DictioEntry.BILINGUAL,
        frenchLexicon.getWord("bilingue", LexicalCategory.ADJECTIVE));
    frenchSensesMapping.put(DictioEntry.CROWD,
        frenchLexicon.getWord("foule", LexicalCategory.NOUN));
    frenchSensesMapping.put(DictioEntry.DEFINITE_DETERMINER,
        frenchLexicon.getWord("le", LexicalCategory.DETERMINER));
    frenchSensesMapping.put(DictioEntry.GREETING_VERB,
        frenchLexicon.getWord("saluer", LexicalCategory.VERB));
    frenchSensesMapping.put(DictioEntry.LANGUAGE_PREPOSITION,
        frenchLexicon.getWord("en", LexicalCategory.PREPOSITION));
    frenchSensesMapping.put(DictioEntry.OLD,
        frenchLexicon.getWord("vieux", LexicalCategory.ADJECTIVE));
    frenchSensesMapping.put(DictioEntry.SPANISH_LANGUAGE,
        frenchLexicon.getWord("espagnol", LexicalCategory.NOUN));
    frenchSensesMapping.put(DictioEntry.YET,
        frenchLexicon.getWord("encore", LexicalCategory.ADVERB));

    return frenchSensesMapping;
}
```

Exemple de génération bilingue

```
public static SPhraseSpec buildGreetCrowd(NLGFactory factory,
    Map<DictioEntry, WordElement> wordSensesMapping) {

    WordElement definiteDeterminer = wordSensesMapping.get(DictioEntry.DEFINITE_DETERMINER);
    WordElement adultMale = wordSensesMapping.get(DictioEntry.ADULT_MALE);
    WordElement old = wordSensesMapping.get(DictioEntry.OLD);
    WordElement bilingual = wordSensesMapping.get(DictioEntry.BILINGUAL);
    WordElement greetingVerb = wordSensesMapping.get(DictioEntry.GREETING_VERB);
    WordElement crowd = wordSensesMapping.get(DictioEntry.CROWD);
    WordElement yet = wordSensesMapping.get(DictioEntry.YET);
    WordElement in = wordSensesMapping.get(DictioEntry.LANGUAGE_PREPOSITION);
    WordElement spanish = wordSensesMapping.get(DictioEntry.SPANISH_LANGUAGE);

    NPPhraseSpec theMan = factory.createNounPhrase(definiteDeterminer, adultMale);
    theMan.addModifier(old);
    theMan.addModifier(bilingual);

    NPPhraseSpec theCrowd = factory.createNounPhrase(definiteDeterminer, crowd);
    theCrowd.setFeature(Feature.PRONOMINAL, true);

    SPhraseSpec greeting = factory.createClause(theMan, greetingVerb, theCrowd);

    greeting.setFeature(Feature.NEGATED, true);
    greeting.setFeature(Feature.TENSE, Tense.PAST);

    greeting.addPreModifier(yet);

    PPPhraseSpec inSpanish = factory.createPrepositionPhrase(in, spanish);
    greeting.addModifier(inSpanish);

    return greeting;
}
```

Exemple de génération bilingue

Sortie :

Bilingual greeting

The old, bilingual man did not yet greet it in Spanish. Le vieil homme bilingue ne l'a pas encore saluée en espagnol.

Conclusion

Bilan

- Réalisation bilingue anglais/français
- Couvre en grande partie la grammaire du *Français fondamental (1^{er} Degré)*
- Lexique français de 3871 entrées
- Documentation française détaillée
- Interface programmeur presque identique
- Fonctionnement interne :
 - 22 classes modifiées
 - 68 méthodes modifiées et 54 méthodes ajoutées
 - 30 classes ajoutées
 - 10 nouveaux packages
 - 29 nouveaux traits (*features*)

Travaux futurs

- Terminer la couverture complète du *Français fondamental (1^{er} Degré)*
 - Phrases interrogatives et propositions relatives
 - Modes conditionnel et subjonctif
 - Noms de nombres, adjectifs numéraux et cardinaux
- Réalisation à partir d'un fichier XML
- Ajouter d'autres langues...
 - Le code est maintenant mieux organisé pour en accueillir de nouvelles