Xerox Incremental Parser

XIP English Grammar

User's Guide

XEROX®

Research Centre Europe

Authored by:

C. Hagege (Categories & Features, Chunk names, Dependencies)

C. Brun (Named Entity Recognition)

XEROX RESEARCH CENTRE EUROPE

6 CHEMIN DE MAUPERTUIS
38240 MEYLAN
FRANCE

Table of Contents

Introduction

One aim of this document is to explain category and chunk names that are most widely used in the XIP English Grammar.

Chunks are basic syntactic domains that are computed by the English grammar and that serve as a basis for the dependency extraction.

Another aim of this document is to list the dependencies that are extracted by the XIP English Grammar. A XIP dependency is a n-ary relation (unary, binary) that binds one or two nodes together from the chunk tree. Usually, dependencies apply on lexical nodes. Each dependency is named with a specific label. A dependency label denotes the grammatical relation that has been computed between the nodes that have been bound together.

Finally, this document describes Named Entity Recognition as part of the XIP English Grammar. Named Entity Recognition aims to "semantically" detect and categorize proper nouns (e.g., "*China*", "*Democratic Party of Russia*"), percentages, money expressions and date expressions.

1 Categories & Features

A category in a XIP grammar must be declared as a set of features (attribute-value pairs).

The PoS (part-of-speech) features are distinguished from other features.

Lexical categories and feature names do not display on screen in the default output mode. However, in the XIP API all the linguistic information (e.g., categories and features) is available within the XipResult object.

1.1 Part-of-Speech Features

NOUN:+	noun
VERB:+	verb
AUX:+	auxiliary
ADJ:+	adjective
NADJ:+	noun-adjective (for ambiguous noun/adjectives)
PREP:+	preposition
ADV:+	adverb
CONJ:+	conjunction
NUM:+	numeral
DIG:+	digit
DET:+	determiner
PRON:+	pronoun
PUNCT:+	punctuation symbol
QUANT:+	quantifier
PART:+	possessive particle
MEAS:+	measure unit
INTERJ:+	interjection
INFTO:+	the infinitive "to" as distinct from a preposition

1.2 Other Features

Features can appear together with Part-of-Speech tags when appropriate.

Here follows the list of the most important features used in the English grammar.

SG:+	singular
PL:+	plural
SP:+	singular-or-plural
MASC:+	masculine
FEM:+	feminine
P1:+	first person ("I" or "We")
P2:+	second person ("You")
P3:+	third person (He/She/It/They)
NOM:+	nominative case (pronouns)
OBL:+	oblique case (pronouns)
GEN:+	genitive
POSS:+	possessive
REFL:+	reflexive
REL:+	relative
PRONPERS:+	personal pronoun
ACRON:+	acronym
PROP:+	proper (for proper names)
INF:+	infinitive
PARTPAS:+	past participle
PAST:+	past tense
PROG:+	progressive form
TRANS:+	transitive
DEF:+	definite
INDEF:+	indefinite
WH:+	wh-pronoun or adverb (e.g. which, when, why)

COORD:+	coordination
SUB:+	subordination
DEC:+	decimal (for numbers)
ROM:+	roman (for numbers)
CARD:+	cardinal
ORD:+	ordinal
PAREN:+	parenthesis
QUOTE:+	quote
SENT:+	end of sentence
COMMA:+	comma
ABBR:+	abbreviation

2 Chunk Nomenclature

Chunks are basic syntactic domains that are computed by the English grammar and that serve as a basis for the dependency extraction.

Here follows the list of chunks used in the XIP English Grammar.

IV	infinitive verbal chunk
FV	finite verbal chunk
GV	gerund verbal chunk
NFV	non-finite verbal chunk
AP	adjectival chunk
NP	nominal chunk
PP	prepositional chunk
ADVP	adverbial chunk
NUMP	numeral chunk
BG	begin of clause chunk (e.g. conjunctions that introduce an embedded clause)
INS	inserted chunk (e.g. comment clauses, parenthesis)
SC	sentential chunk (from the beginning of a clause until the first finite verb)

3 Dependency Nomenclature

A XIP dependency is a n-ary relation (unary, binary) that binds one or two nodes together from the chunk tree. Usually, dependencies apply on lexical nodes. Each dependency is named with a specific label. A dependency label denotes the grammatical relation that has been computed between the nodes that have been bound together.

AGENT

This binary dependency links the inflected auxiliary of a passive verbal form to its agent complement

Example
The mouse was eaten by the cat

AGENT(was,cat)

COMPAR

This dependency links a “than” (comparative) to a following adjective

Example
He is more vigorous than nice

COMPAR(than,nice)

COMPOUND

This dependency links a form of the verb “have” and “got”

Example
I have got a friend.

COMPOUND(have,got)

CONNECTOR_COMPLTHAT

This dependency links the verb of a that-clause to the “that” introducing this clause.

Example
They said that the investigation succeeded.

CONNECTOR_COMPLTHAT(succeeded,that)

CONNECTOR_SUBORD

It links the verb of a subordinated clause (that is neither a relative, nor a that-clause) to the subordinated conjunction introducing this clause.

Example

I will stay at home because it is raining.

CONNECTOR_SUBORD(is,because)

COORD

This binary relation links a coordination conjunction to one of the coordinated elements. **All** the coordinated elements of a coordinated expression have to be linked by a COORD dependency.

Example
He ate, played and drank.

COORD(and,ate)
COORD(and,played)
COORD(and,drank)

COORD_NEG

This binary relation links a coordination conjunction to one **negated coordinated element**.

Example
Pastels contain no cadmium or other pigments.

COORD_NEG(or,cadmium)

DETD

This binary relation links a nominal head and a determiner

Examples
These three girls

DETD(girls,These)

On the table

DETD(table,the)

HEAD

This is a binary relation relating the nucleus of some chunk with the chunk itself.

Examples
The three girls

HEAD(girls,The three girls)

Really nice

HEAD(nice,Really nice)

On the table

HEAD(table,On the table)

IOBJ_POST

This dependency links a verb to an indirect object (that is not introduced by any preposition) on its right.

Example
He gave Mary some flowers.

IOBJ_POST(gave,Mary)

The MAIN suite*

MAIN* is a list of unary dependencies that marks the main verbal element of the principal clause of the parsed sentence. It corresponds to the last element of the verbal chain.
A list of suffixes added to the MAIN dependencies is available in order to type aspectual and temporal features of the main verb.
Here follows the detailed list of the possible suffixes that can appear with a MAIN dependency. All the different suffixes can be combinde together.

_INSIST

This suffix is present on the MAIN dependency when the main verb is preceded by a DO-auxiliary.

Example
They do like French films.

MAIN_INSIST(like)

_MODAL

This suffix is present on the MAIN dependency when the main verb is preceded by a modal auxiliary.

Example
They should come.

MAIN_MODAL(come)

_PASSIVE

This suffix is present on the MAIN dependency when the main verb is in the passive form.

Example
The mouse was eaten by the cat.

MAIN_PASSIVE(eaten)

_PERFECT

This suffix is present on the MAIN dependency when the main verb is in perfect.

Example
They have eaten the cake.

MAIN_PERFECT(eaten)

_PROGRESS

This suffix is present when the main verb is in the progressive form

Example
When are you going?

MAIN_PROGRESS(going)

As we mentioned before, suffixes can combine, so, for a sentence like:
He should have been caught
We obtain the following MAIN dependency:

MAIN_MODAL_PERFECT_PASSIVE(caught)

IMPORTANT NOTE: If the main verbs of a sentence are coordinated, the MAIN dependency marks the coordinator.

Example
Merrymakers tossed confettis, squawked horns and popped champagne corks.

MAIN(and).

The MOD suite*

This set of dependencies links a governor and any kind of complements or adjuncts attached to this governor. Note that the governor can be a verb, a noun, an adjective or an adverb. Note also that we do not make any distinction between complements and adjuncts.

To the MOD dependencies we add three kinds of suffixes:

- position suffixes
- syntactic suffixes
- typing suffixes

Position suffixes

They express the relative position of the modifier considering the governor is modified.

_PRE

This suffix expresses that the modifier is on the left of the governor in the sentence.

Examples

They develop human overhead capital.

MOD_PRE(capital,overhead)
MOD_PRE(capital,human)

He always sleeps.

MOD_PRE(sleeps,always)

_POST

This suffix expresses that the modifier is on the right of the governor in the sentence.

Examples

The new version will combine index with customized services.

MOD_POST(combine,services)

Syntactic suffixes

These suffixes express the syntactic nature of the modifier when the modifier is not nominal. They are always preceded by a position suffix (except the _NEG suffix, see below). Here follows the list of these suffixes:

_GERUND

This suffix expresses the fact that the modifier is an ING-form.

Example
He left, after having done the job.

MOD_POST_GERUND(left,done)

_NEG

This suffix expresses the fact that the modifier has a negative polarity.

Example
They do not work.

MOD_NEG(work,not)

_SENTENCE_RELATIV

This suffix expresses that the modifier is a relative clause. In this case, the modifying clause is represented by its main verb.

Example
The book that I'm reading is interesting.

MOD_POST_SENTENCE_RELATIV(book,reading)

_SENTENCE

This suffix expresses that the modifier is sentential (excluding the case of relatives, see above). Like with relative clauses, the sentential modifier is represented by its main verb.

Example
She left because he arrived.

MOD_POST_SENTENCE(left,arrived)

_PROPER

This suffix is used in chains of complex proper nouns in order to link the last name to other elements of the chain.

Example
John W. Smith left.

MOD_PROPER(Smith,John)
MOD_PROPER(Smith,W.)

Typing suffixes

They correspond to a first attempt to type a semantic nature of modifiers.
This typing is not exhaustively implemented in the current grammar version.

Here is the list of these suffixes.

_AGENTIF

It is added to the link between a modifying ING-form and a nominal head, when this nominal head is the agent of the action expressed by the ING-form

Example
Recessive host mutation affecting virus multiplication was discussed in this paper.
MOD_POST_AGENTIF(mutation,affecting)

_DURATION

This suffix types a modifier link, when the modifier expresses an idea of temporal duration.

Example
He waited for hours.

MOD_POST_DURATION(waited,hours)

_LOC

This suffix types a modifier link, when the modifier expresses an idea of spatial location.

Example
He stayed in Paris.

MOD_POST_LOC(stayed,Paris)

_MANNER

This suffix types a modifier link when the modifier is a manner adverb.

Example
She spoke gently.

MOD_POST_MANNER(spoke,gently)

_QUANTITY

This suffix is added to specify that the modifier expresses a quantity. This is done only when the modifier is a WH-expression.

Example
How much did he pay?

MOD_QUANTITY(pay,how much)

_TEMP

This suffix is added to MOD* dependency when the modifier is a temporal expression

Example
He left yesterday.

MOD_POST_TEMP(left,yesterday)

The NUCL suite*

This is a binary dependency that links elements of the verbal chain.

Different suffixes are available for this dependency.

_VLINK

VLINK is the suffix that is added on a NUCL dependency when the two linked lexical units are elements of the same complex verbal chain.

To the _VLINK suffix, the suffixes available for the MAIN* suite can also be added.

Examples
Caldwell's resignation had been expected for some time.

NUCL_VLINK_PERFECT(had,been)
NUCL_VLINK_PASSIVE(been,expected)

_PARTICLE

PARTICLE is the suffix that is added on VLINK to express the relation between a verbal form and a verbal particle.

Example
The Georgia legislature will wind up its 1961 session Monday.

NUCL_PARTICLE(wind,up)

_SUBJCOMPL

NUCL_SUBJCOMPL links a copula and its complement

Example
Despite the warning, there was a unanimous vote to enter a candidate, according to Republicans who attended.

NUCL_SUBJCOMPL(was,vote)

*OBJ**

OBJ* is a binary relation between the last element of the verbal chain and its direct object.

When the object is on the left of the verb we have the dependency OBJ_PRE.
When the object is on the right of the verb, we have the dependency OBJ_POST.

Examples
I gave Mary some flowers.

OBJ_POST(gave,flowers)

What did you give her?

OBJ_PRE(give,What)

Other suffixes can be added to this dependency. Here follows the list:

_SENTENCE

It marks that the object is sentential. It is represented by the verb of the object clause.

Example
He said that she would come.

OBJ_POST_SENTENCE(said,come)

_GERUND

It marks that the object is a verbal ING-form.

Example
He went fishing.

OBJ_POST_GERUND(went,fishing)

_INFINIT

This suffix expresses the fact that the object is an infinitive form.

Example
I like to play.

OBJ_POST_INFINIT(like,play)

_NEG

This suffix expresses the fact that the object is negated.

Example
They have no choice.

OBJ_POST_NEG(have,choice)

_RELATIV

This suffix expresses the fact that the object is a relative pronoun that has an object function within the relative clause.

Example
They have a book that I like to read.

OBJ_PRE_RELATIV(like,that)

OBJCOMPL_POST

This relation links a verb with an object complement

Example
They elected him president.

OBJCOMPL_*POST*(elected,president)

PREPD

This binary relation links a preposition to the nominal head of a PP.

Example
The book is on the table.

PREPD(book,on)

QUANTD

This binary relation links a quantifier and a nominal head.

Examples
Some men arrived

QUANTD(men,Some)

We invited hundreds of people

QUANTD(people,hundreds)

*SUBJ**

SUBJ* is a binary relation between the inflected element of the verbal chain and its subject.

When the subject is on the left of the verb we have the dependency SUBJ_PRE.
When the subject is on the right of the verb, we have the dependency SUBJ_POST.

Together with the position suffixes, we have a list of other suffixes added to the SUBJ dependency.

_GERUND

This suffix is used when the subject is a ING-form

Example
Going to the cinema is nice.

SUBJ_PRE_GERUND(is,Going)

_NFINIT

This suffix expresses the fact that the verb involved in this SUBJ dependency is an infinitive.

Example
I saw her come.

SUBJ_PRE_NFINIT(come,her)

_NEG

This suffix is added to the SUBJ *dependency* when the subject is negated.

Example
No books are left.

SUBJ_PRE_NEG(are,books)

4 Named Entity Recognition

This describes the list of dependencies and features used for Named Entity Recognition as part of the XIP English Grammar.

Named Entity Recognition aims to "semantically" detect and categorize proper nouns, which can be single nouns like "*China*", or names like "*Democratic Party of Russia*". In addition, XIP recognizes percentages, money expressions and date expressions.

Named entities are output as unary dependencies. Moreover, some features involved for Named Entity Recognition can be output as well, that can give more fine-grained Named Entity Recognition (e.g. distinguish first name and full name of a person).

The different types of named entities that can currently be recognized are the following:

- Percentages: *10 % , 10 percent*
- Dates: *March 4, 1991*
- Monetary expressions: *$26 billion*
- Locations: *San Francisco, Mount Everest, West Bank*
- Person Names: *President George W. Bush, Kofi Annan, Edward III of England*
- Organizations: *British Airways, Bang & Olufsen, Bank of Brazil*
- Events: *World War II, America's Cup*
- Legal references: *Warsaw Pact, Maastricht Treaty*

4.1 Dependencies That Characterize Named Entities

PERCENT

This unary dependency characterizes a percentage expression.

Example

He gave 10 per cent of his salary to charities.

PERCENT(10 per cent)

TEMPEXPR

This unary dependency characterizes temporal expressions.

Example

..., because of currency crises from time to time in Latin America.

TEMPEXPR(from time to time)

TEMPEXPR_DATE

This unary dependency characterizes temporal expressions which are dates.

Example

The December 1996 acquisition of Proler International increased scrap collection to about 2.5 million tons.

TEMPEXPR_DATE(December 1996)

MONEY

This unary dependency characterizes monetary expressions.

Example

This company has annualized revenues of approximately $170 million.

MONEY($170 million)

LOCATION

This unary dependency characterizes places.

Example

It is the 50th Anniversary of the First Ascent of Mount Everest

LOCATION(Mount Everest)

LOCATION_COUNTRY

This unary dependency characterizes places which are countries.

Example

Its location is in the United States.

LOCATION_COUNTRY(United States)

LOCATION_CITY

This unary dependency characterizes places which are cities.

Example

He lives in Buenos Aires.

LOCATION_CITY(Buenos Aires)

PERSON

This unary dependency characterizes person names.

Example

King Henry VIII split with the Roman Catholic Church over a question of his divorce from Catherine of Aragon.

PERSON(King Henry VIII)
PERSON(Catherine of Aragon)

ORGANIZATION

This unary dependency characterizes organization names.

Example

We believe there is unusual investment potential in equities of scrap processors such as Philip Services Corp. and Metal Management Inc.

ORGANIZATION(Philip Services Corp.)
ORGANIZATION(Metal Management Inc.)

EVENT

This unary dependency characterizes events.

Example

Memorial will be the first national memorial dedicated to all who served during World War II.

EVENT(World War II)

LAW

This unary dependency characterizes law related names.

Example

The Treaty of Troyes reduced Valois rule to the lands south of the Loire River.

LAW(Treaty of Troyes)

4.2 Features Related to Named Entities

Percentage related features:

percent:+

Date related features:

Main Feature:	Secondary Features:
date:+	hour:+
	day:+
	month:+
	year:+
	shortyear:+
	period:+
	tempexpr:+

Money related features:

Main Feature:	Secondary Features:
money:+	curr:+
	guesscurr:+

Location related features:

Main Feature:	Secondary Features:
place:+	city:+
	country:+
	address:+
	modloc:+
	locpost:+
	locpre:+

Person related features:

Main Feature:	Secondary Features:
person:+	firstname:+
	title:+
	prof:+
	famlink:+
	celeb:+
	fullname:+
	particlename:+

Organization related features:

Main Feature:	Secondary Features:
org:+	orgcountry:+
	orghead:+
	orgmod:+
	orgend:+
	bus:+

Event related features:

Main Feature:	Secondary Features:
event:+	eventmod:+

Law related features:

Main Feature:	Secondary Features:
law:+	lawmod:+