[Every possum ]x was brown.

[Every possum ]x x was brown.

A x possum(x) brown(x)

A x possum(x) --> brown(x)

====================================

John ate [a sandwich]x

[a sandwich]x John ate x

E x sandwich(x) eat(j, x)

E x sandwich(x) & eat(j, x)

=========================================

[A young woman]x was speaking.

[A young woman]x was speaking.

[A young woman]speak(x)

E x woman(x) & young(x)speak(x)

E x [woman(x) & young(x) &speak(x)]

============================================

Kerry filled [all the gaps]

[all the gaps] Kerry filled x

[all the gaps] fill(k, x)

A x gap(x) fill(k, x)

A x gap(x) --> fill(k, x)

Every guest thanked Jones

A x gap(x) --> fill(k, x)

2a. There was a black hat on the bed

E x[ hat(x) &black(x) & on(x, the bed)]

2 All roads lead to Rome.

[All roads]x x leads to Rom

A x road(x) lead-to(x, Rome)

A x[ road(x) --> lead-to(x, Rome)]

A x [(road(x) → lead-to(x, Rome)]

Utopia welcomes all travelers from Spain

Utopia welcomes [all travelers from Spain]

[all travelers from Spain] Utopia welcomes x

Ax (traveler(x) & from(x, Spain)) welcome(Utopia, x)

Ax [(traveler(x) & from(x, Spain)) --> welcome(Utopia, x)]

~~Ax [traveler(x) --> from(x, Spain) & welcome(Utopia, x)]~~

Clive got murdered. <===> Some one murdered Clive.

[Some one] murdered Clive.

[Some one] x murdered Clive.

E x murder(x, Clive)

Jones read every book in the library.

Jones read [every book in the library]

[every book in the library] Jones read x

[every book in the library] read(j,x)

A x (book(x) & in( x , the library) ) read(j,x)

A x (book(x) & in( x , the library) ) --> read(j,x)

============

Reading 1

Clive gave [every child ] [ a biscuit or batman-comic]

Clive gave [every child ]y [ a biscuit or batman-comic]x

\*\* [every child ]y [ a biscuit or batman-comic]x Clive gave x to y

[every child ]y [ a biscuit or batman-comic]x give(C,x,y)

[every child ]y Ex biscuit(x) \/ batman-comic(x) give(C,x,y)

A y [child(y) --> (Ex (biscuit(x) \/ batman-comic(x) ) & give(C,x,y)) ]

========

Reading 2

\*\* [ a biscuit or batman-comic]x [every child ]y Clive gave x to y

Ex (biscuit(x) \/ batman-comic(x) ) & A y [child(y) --> give(C,x,y)) ]

========

There's no business like show business.

show-business(x) ==> treating show business as a predicate!!!

show business ==> sb treating show business as a roper noun

no ==> ~ Ex

~ Ex business(x) & "x is like show business"

~ Ex [business(x) & like(x, sb)]