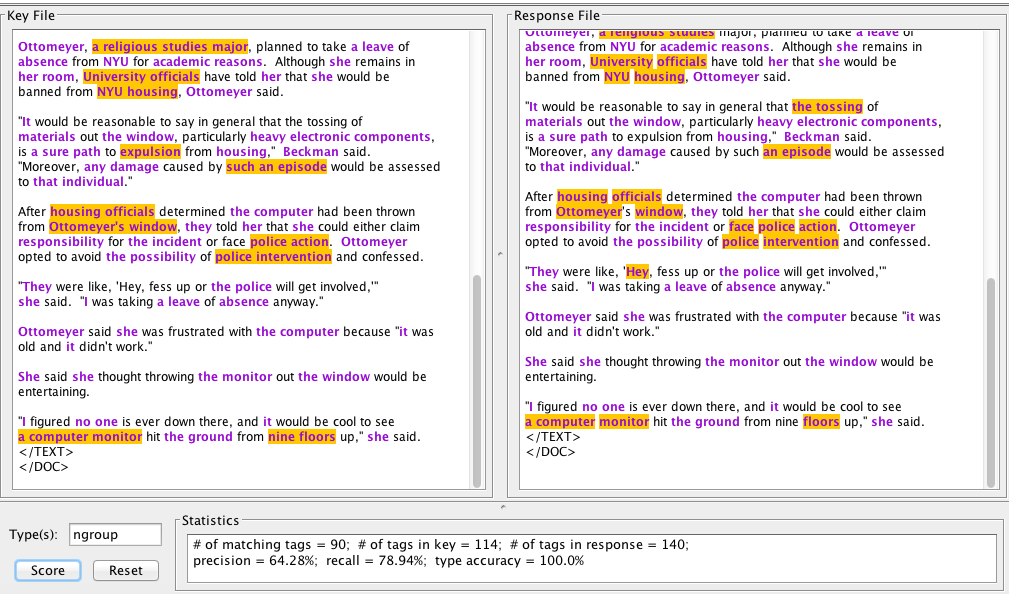
Ans1.

(i)

# of matching tags = 90; # of tags in key = 114; # of tags in response = 140;

precision = 64.28%; recall = 78.94%; type accuracy = 100.0%

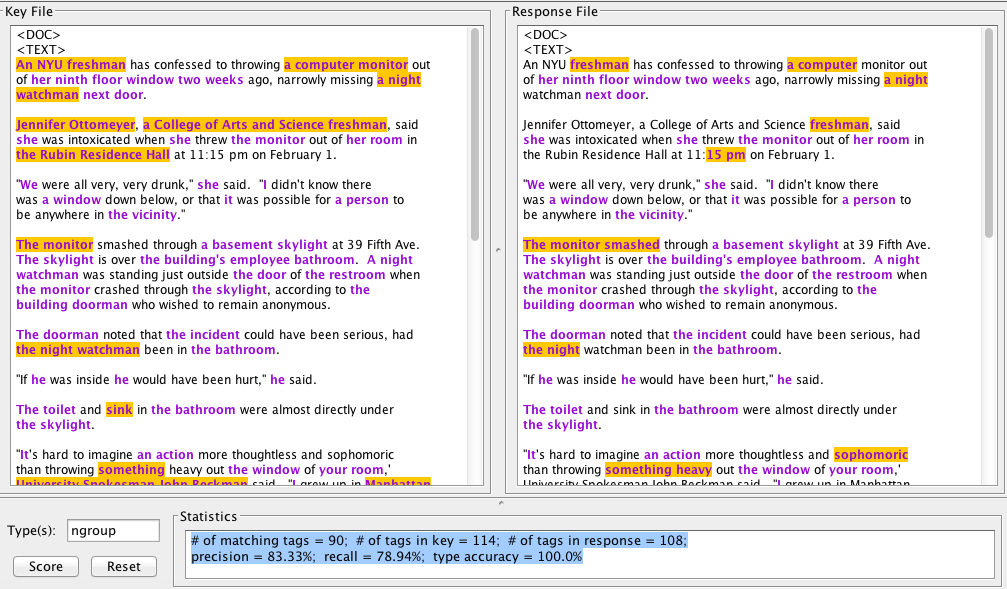


(ii)

For Assignment 4 Grammar:

# of matching tags = 90; # of tags in key = 114; # of tags in response = 108;

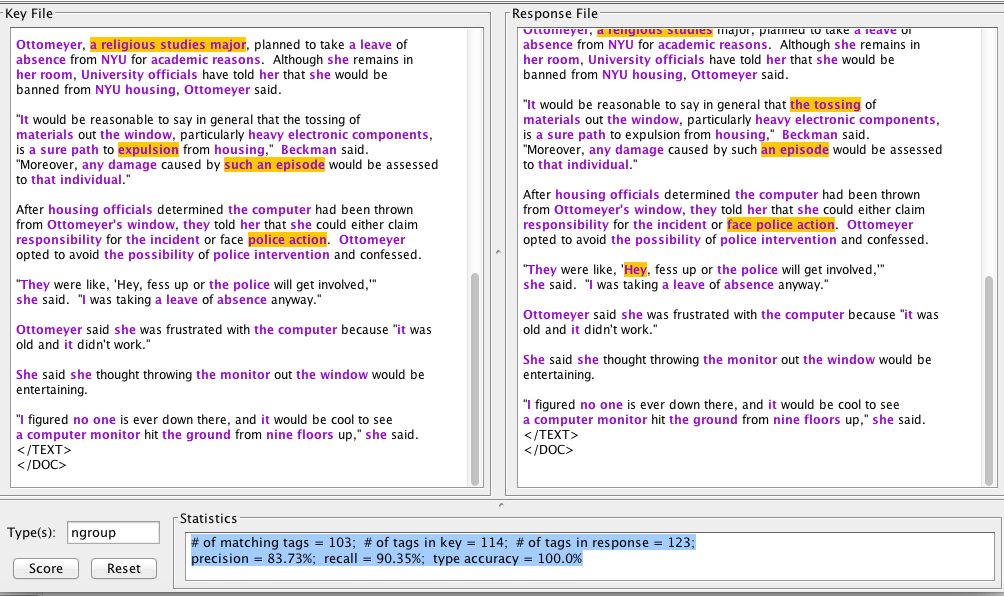
precision = 83.33%; recall = 78.94%; type accuracy = 100.0%



After Improvement:

# of matching tags = 103; # of tags in key = 114; # of tags in response = 123;

precision = 83.73%; recall = 90.35%; type accuracy = 100.0%

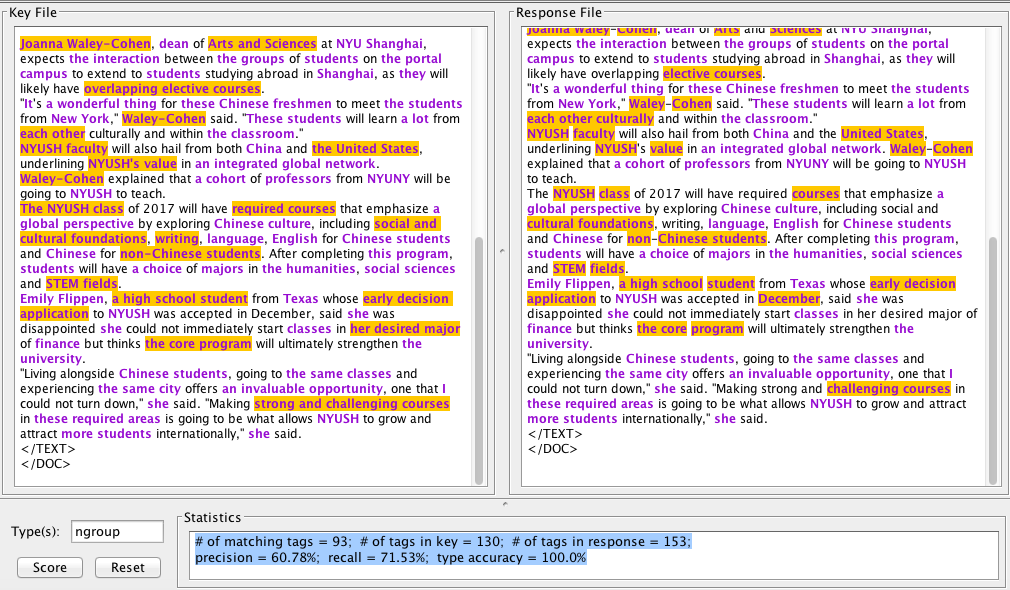


(iii) For article 2

Original Grammar

# of matching tags = 93; # of tags in key = 130; # of tags in response = 153;

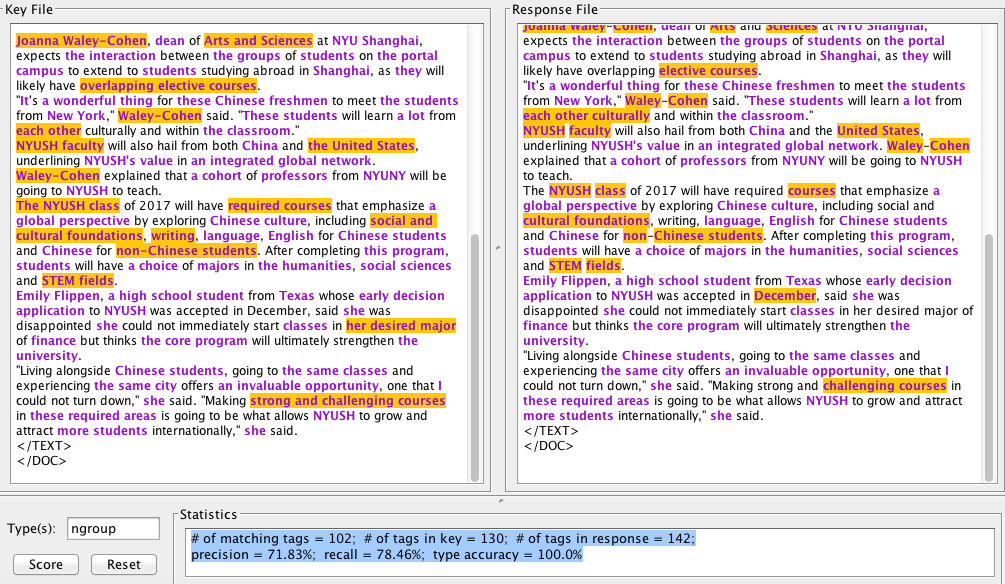
precision = 60.78%; recall = 71.53%; type accuracy = 100.0%



Assignment 4 Grammar

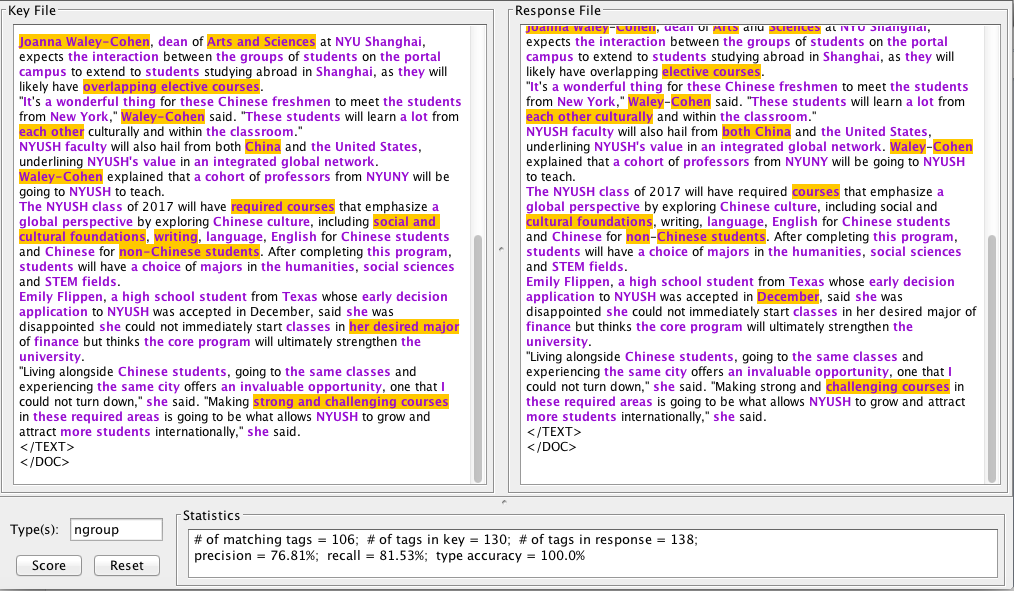
# of matching tags = 102; # of tags in key = 130; # of tags in response = 142;

precision = 71.83%; recall = 78.46%; type accuracy = 100.0%



After Improvement

# of matching tags = 106; # of tags in key = 130; # of tags in response = 138; precision = 76.81%; recall = 81.53%; type accuracy = 100.0%

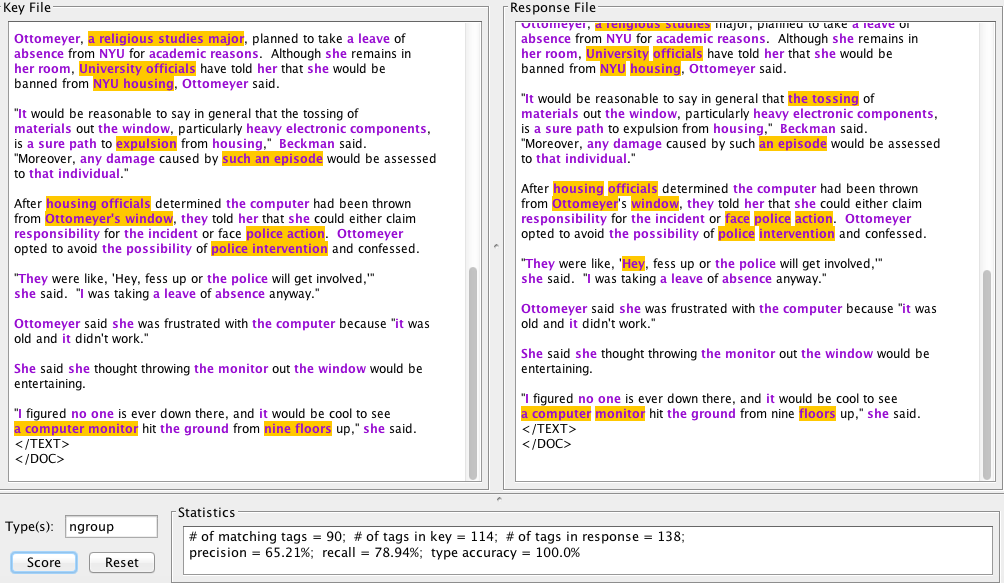


Ans2.

Initial Grammar

# of matching tags = 90; # of tags in key = 114; # of tags in response = 138;

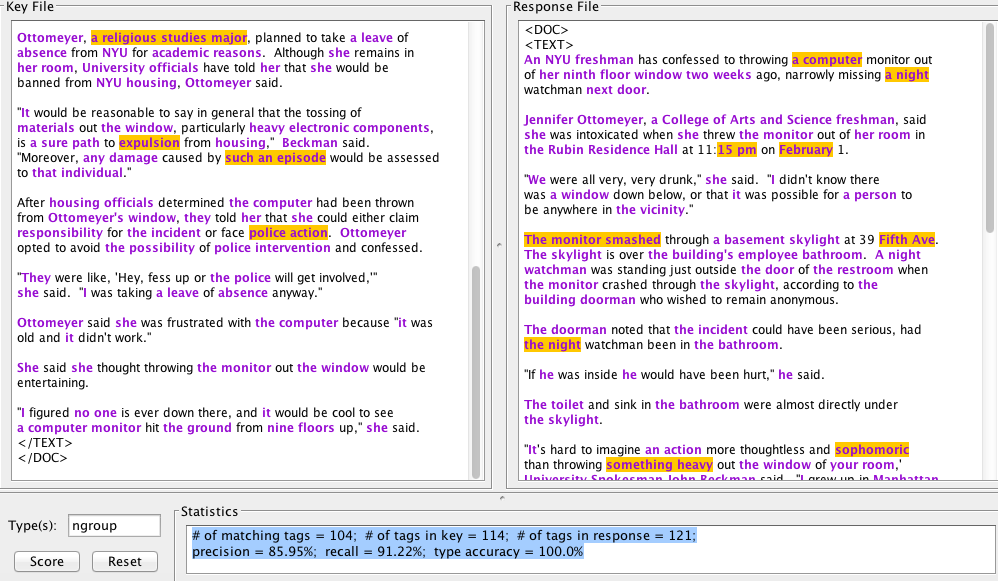
precision = 65.21%; recall = 78.94%; type accuracy = 100.0%



Final Chunk Patterns:

# of matching tags = 104; # of tags in key = 114; # of tags in response = 121;

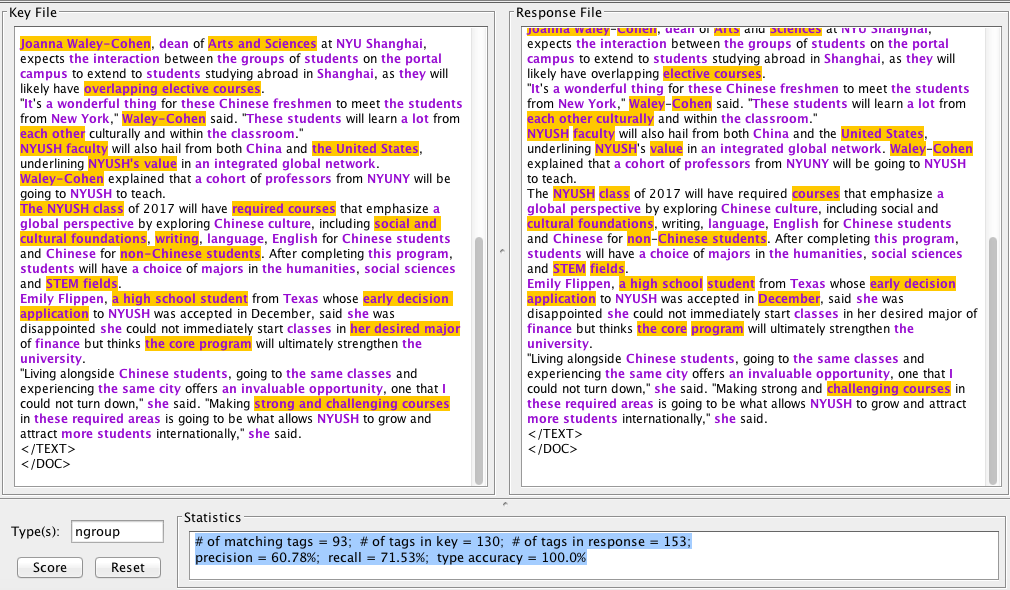
precision = 85.95%; recall = 91.22%; type accuracy = 100.0%



Original Grammar (ARTICLE -2)

# of matching tags = 96; # of tags in key = 130; # of tags in response = 150;

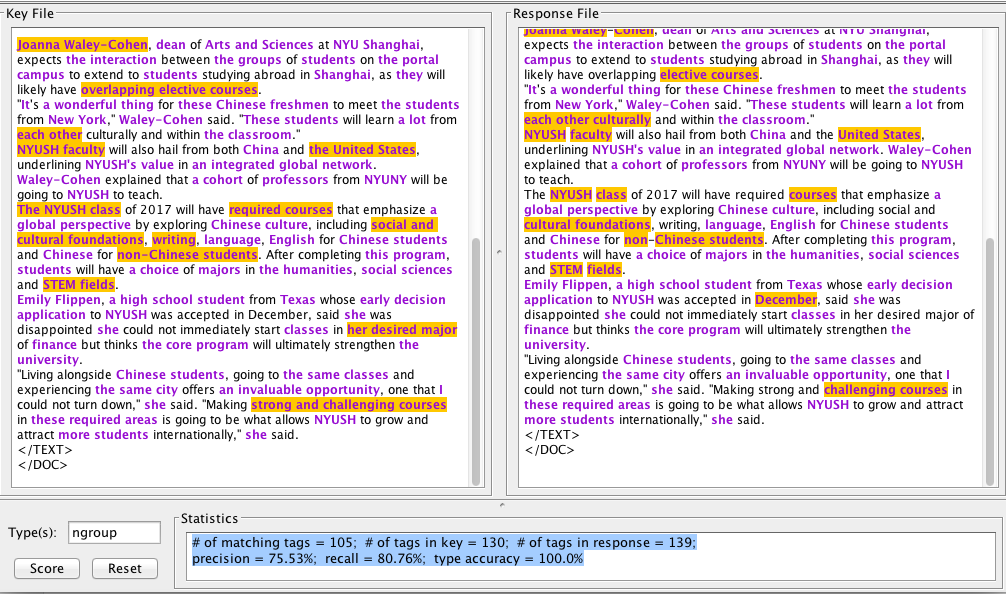
precision = 64.0%; recall = 73.84%; type accuracy = 100.0%



Final Grammar(ARTICLE-2)

# of matching tags = 105; # of tags in key = 130; # of tags in response = 139;

precision = 75.53%; recall = 80.76%; type accuracy = 100.0%



FINAL GRAMMAR

pattern set chunks;

// patterns for noun groups

ng := det-pos? [constit cat=adj]\* [constit cat=n]+ |

qu [constit cat=adj]\* [constit cat=n]+ |

proper-noun |

[constit cat=pro] ;

qu := [constit cat=det]? [constit cat = q];

det-pos := [constit cat=det] |

[constit cat=det]? [constit cat=n number=singular] "'s" |

[constit cat=det]? [constit cat=adj]\* proper-noun "'s" ;

proper-noun := ([token case=cap ] | [undefinedCap])+ | [ENAMEX]+ ;

when ng add [ngroup];

//Pattern for acive vgroups

vg := [constit cat=tv] |

tv-vbe vg-ving |

tv-vb vg-ven |

[constit cat=w] [constit cat=v] |

[constit cat=w] tv-vbp vg-ven |

[constit cat=w] vg-inf ;

vg-inf := [constit cat=v] |

"be" vg-ving;

vg-ven := [constit cat=ven] |

"been" vg-ving ;

vg-ving := [constit cat=ving] ;

tv-vbe := "is" | "are" | "was" | "were";

tv-vb := tv-vbp | tv-vbs;

tv-vbp := "have";

tv-vbs := "has" | "had";

con := "of" | "and" | "for";

when vg add [constit cat=vgroup];

// patterns for passive verb groups

vg-pass := tv-vbe [constit cat=ven] |

tv-vb "been" [constit cat=ven] |

tv-vbs "been" "being" [constit cat=ven] |

[constit cat=w] "be" [constit cat=ven] |

[constit cat=w] tv-vbp "been" [constit cat=ven] |

[constit cat=w] "be" "being" [constit cat=ven] |

[constit cat=w] tv-vbp "been" "being" [constit cat=ven] |

tv-vbe "being" [constit cat=ven];

when vg-pass add [constit cat=vgroup-pass];

// pattern for infinitival verb groups

to-vg := vg-inf;

when to-vg add [constit cat=vgroup-inf];