# Colors

# Example to add "COLOR" as ner tag, and hex RGB code as the normalized tag for strings matching a color

# Case insensitive pattern matching (see java.util.regex.Pattern flags)

ENV.defaultStringPatternFlags = 2

# Map variable names to annotation keys

ner = { type: "CLASS", value: "edu.stanford.nlp.ling.CoreAnnotations$NamedEntityTagAnnotation" }

normalized = { type: "CLASS", value: "edu.stanford.nlp.ling.CoreAnnotations$NormalizedNamedEntityTagAnnotation" }

tokens = { type: "CLASS", value: "edu.stanford.nlp.ling.CoreAnnotations$TokensAnnotation" }

# Create OR pattern of

# regular expression over tokens to hex RGB code

# for colors and save it in a variable

$Colors = (

/red/ => "#FF0000" |

/green/ => "#00FF00" |

/blue/ => "#0000FF" |

/magenta/ => "#FF00FF" |

/cyan/ => "#00FFFF" |

/orange/ => "#FF7F00" |

/brown/ => "#964B00" |

/purple/ => "#800080" |

/gray/ => "#777777" |

/black/ => "#000000" |

/white/ => "#FFFFFF" |

(/pale|light/) /blue/ => "#ADD8E6"

)

# Define ruleType to be over tokens

ENV.defaults["ruleType"] = "tokens"

# Define rule that

# upon matching pattern defined by $Color

# annotate matched tokens ($0) with ner="COLOR" and normalized=matched value ($$0.value)

{ pattern: ( $Colors ),

action: ( Annotate($0, ner, "COLOR"), Annotate($0, normalized, $$0.value ) ) }