Plots Research Project

Maximilian Pfundstein

2020-01-08

Contents

1 Evaluation

This RMarkdown file is just meant for holding the results and plotting them. Most of the textual analysis in done in the report.

2 Cache Speedup

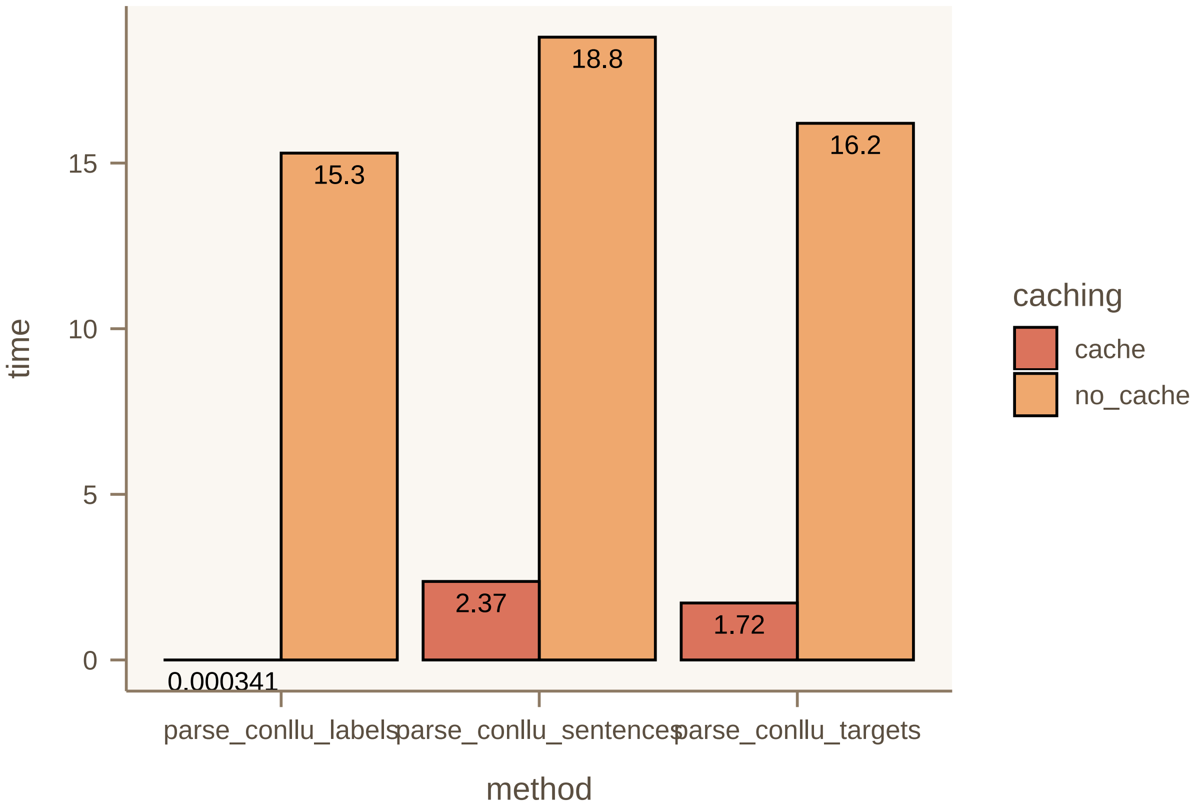
```
cache_speedup = data.frame(
  caching=rep(c("no_cache", "cache"), each=3),
  method=rep(c("parse_conllu_labels", "parse_conllu_sentences", "parse_conllu_targets"), 2),
  time=c(15.3, 18.8, 16.2, 341 * 10^(-6), 2.37, 1.72))

head(cache_speedup)
```

```
##    caching                 method     time
## 1 no_cache    parse_conllu_labels 1.53e+01
## 2 no_cache parse_conllu_sentences 1.88e+01
## 3 no_cache   parse_conllu_targets 1.62e+01
## 4    cache    parse_conllu_labels 3.41e-04
## 5    cache parse_conllu_sentences 2.37e+00
## 6    cache   parse_conllu_targets 1.72e+00
```

```
ggthemr('dust', layout="minimal", spacing=2)
```

```
## Warning: New theme missing the following elements: axis.ticks.length.x,
## axis.ticks.length.x.top, axis.ticks.length.x.bottom, axis.ticks.length.y,
## axis.ticks.length.y.left, axis.ticks.length.y.right
```

```
ggplot(data=cache_speedup, aes(x=method, y=time, fill=caching)) +
  geom_bar(stat="identity", color="black", position=position_dodge())+
  geom_text(aes(label=time), vjust=1.6, color="black",
            position = position_dodge(0.9), size=3.5)
```



3 Scores for Multitask Learning

They're seperated as R cannot handle lines that are too long.

3.1 Data

Will float out of the pdf. Maybe I will fix this later.

```
syn_dev_single = c(0.001264955463026406, 0.369744196932566, 0.45900577329802034, 0.6865783671363118, 0.7

syn_dev_single = c(syn_dev_single, 0.9374284421013394, 0.9370841539362325, 0.9381126476977693, 0.9368995

syn_dev_raw = c(0.3230067945299734, 0.7609164942160818, 0.8331008329383084, 0.8707282819501104, 0.882584

syn_dev_raw = c(syn_dev_raw, 0.9432123162378326, 0.9410893511233356, 0.9424691918450179, 0.9422396544184

syn_dev_multi = c(0.05969790527154771, 0.06280650845013422, 0.06315995963587459, 0.0636530240399824, 0.0

syn_dev_multi = c(syn_dev_multi, 0.06514635529973543, 0.06518170041830947, 0.06513928627602064, 0.06517

syn_dev_loop = c(0.06700347237746083, 0.668924800962345, 0.8204022425216607, 0.8124410240070588, 0.8649

syn_dev_loop = c(syn_dev_loop, 0.9413839279795287, 0.9420218499248161, 0.9417915002587339, 0.94223240205

sem_only = c(0.028604612576738013, 0.710129184932304, 0.8136029023282025, 0.8399491902191173, 0.86008209
sem_only = c(sem_only, 0.9257053927870628, 0.9250059063755105, 0.9243974132863022, 0.9237146192498351, 0

sem_dev_single = c(0.0, 0.08400061359104155, 0.2694347033244494, 0.552456976985279, 0.6220755770904509,

sem_dev_single = c(sem_dev_single, 0.923371842502587, 0.9225398431570959, 0.9232545896099044, 0.92335284

sem_dev_raw = c(0.14235071205916508, 0.689131179588464, 0.7872509384926364, 0.8266844985658226, 0.833527

sem_dev_raw = c(sem_dev_raw, 0.9260092744135298, 0.9253422098434156, 0.9254250451636867, 0.925435398260

sem_dev_multi = c(0.6227885290924187, 0.7806938271149554, 0.8326245123194265, 0.849293325002955, 0.86564

sem_dev_multi = c(sem_dev_multi, 0.924836159718938, 0.925313317257073, 0.9248833460592178, 0.9253045432

sem_dev_loop = c(0.009887196081075009, 0.6184216035168515, 0.7687221603563473, 0.8076335073013864, 0.82

sem_dev_loop = c(sem_dev_loop, 0.9233672185768198, 0.9229697587495753, 0.9225327082240239, 0.922362826

sem_only_val = 0.9239
sem_only_ood = 0.875

sem_single_val = 0.926
syn_single_val = 0.9464
sem_single_ood = 0.8741
syn_single_ood = 0.9081

sem_raw_ood = 0.8780
syn_raw_ood = 0.9092

sem_loop_val = 0.9251
syn_loop_val = 0.9429
sem_loop_ood = 0.8787
syn_loop_ood = 0.9096

sem_multi_ood = 0.8771
syn_multi_ood = 0.0739
```

3.2 Plotting

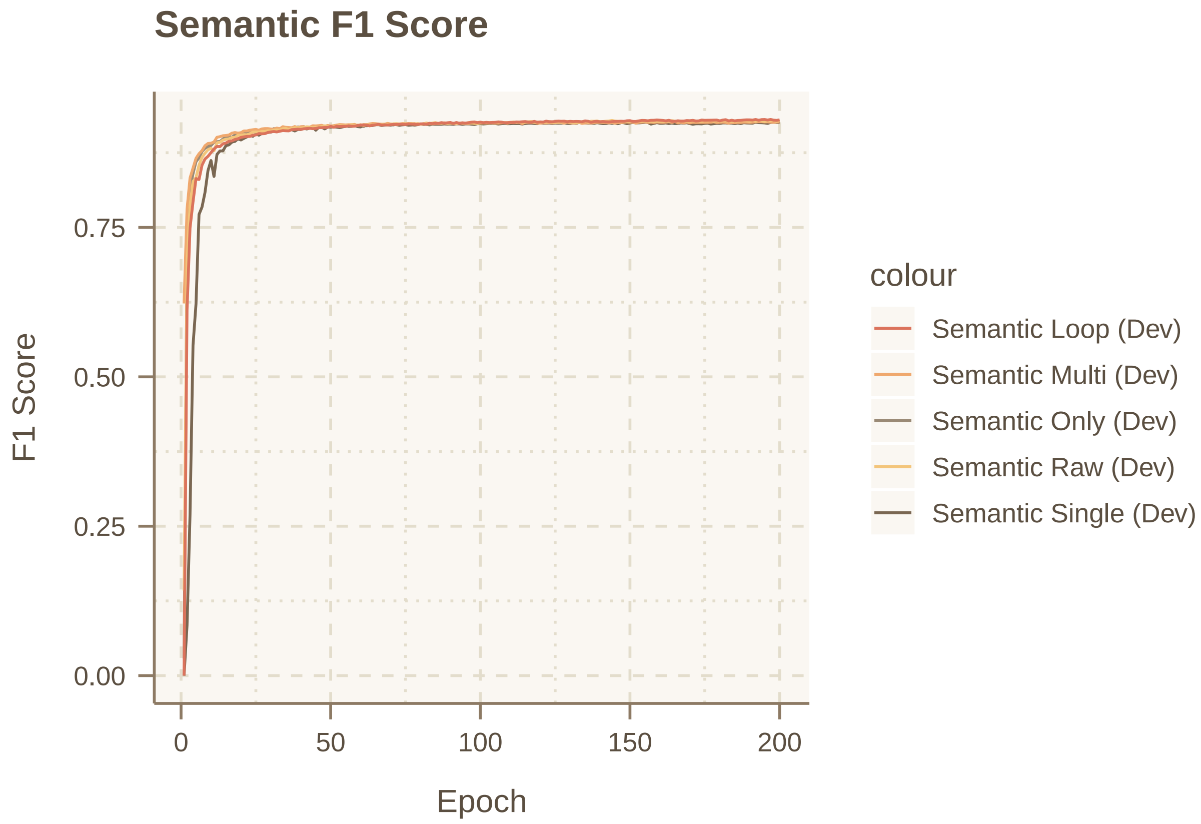
Trimming some data and creating data frame.

```
# Forget to empty old entries when logging, to getting rid of them now
sem_dev_loop = sem_dev_loop[38:237]
syn_dev_loop = syn_dev_loop[38:237]

df = data.frame(index = 1:200, sem_dev_loop, sem_dev_multi, sem_dev_raw, sem_dev_single, sem_only,
                syn_dev_loop, syn_dev_multi, syn_dev_raw, syn_dev_single)
```

```
ggthemr('dust', layout="scientific", spacing=2)
```

```
## Warning: New theme missing the following elements: axis.ticks.length.x,
## axis.ticks.length.x.top, axis.ticks.length.x.bottom, axis.ticks.length.y,
## axis.ticks.length.y.left, axis.ticks.length.y.right
```

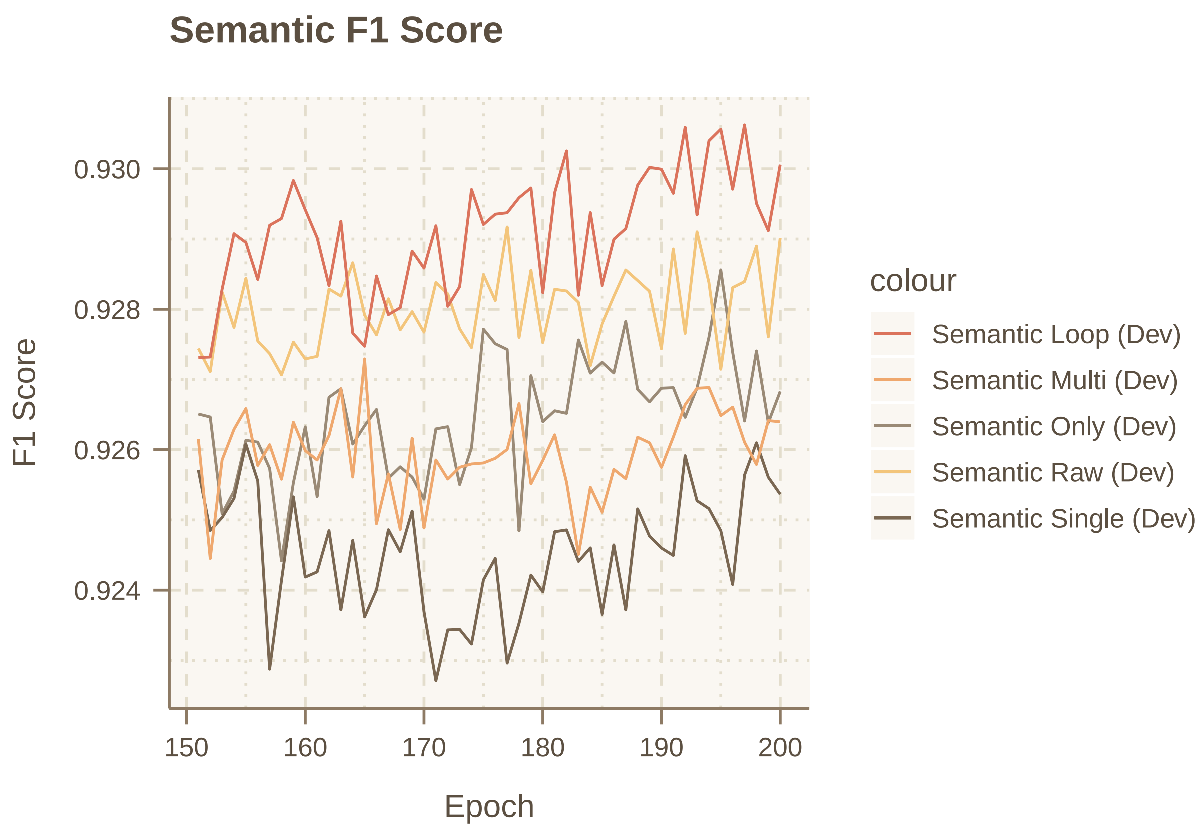
```r
ggplot(df) +
  geom_line(aes(x=index, y=sem_only, color="Semantic Only (Dev)")) +
  geom_line(aes(x=index, y=sem_dev_single, color="Semantic Single (Dev)")) +
  geom_line(aes(x=index, y=sem_dev_multi, color="Semantic Multi (Dev)")) +
  geom_line(aes(x=index, y=sem_dev_raw, color="Semantic Raw (Dev)")) +
  geom_line(aes(x=index, y=sem_dev_loop, color="Semantic Loop (Dev)")) +
  labs(title = "Semantic F1 Score", y = "F1 Score", x = "Epoch")
```



```r
ggthemr('dust', layout="scientific", spacing=2)
```

```
## Warning: New theme missing the following elements: axis.ticks.length.x,
## axis.ticks.length.x.top, axis.ticks.length.x.bottom, axis.ticks.length.y,
## axis.ticks.length.y.left, axis.ticks.length.y.right
```

```r
ggplot(df[151:200,]) +
  geom_line(aes(x=index, y=sem_only, color="Semantic Only (Dev)")) +
  geom_line(aes(x=index, y=sem_dev_single, color="Semantic Single (Dev)")) +
  geom_line(aes(x=index, y=sem_dev_multi, color="Semantic Multi (Dev)")) +
  geom_line(aes(x=index, y=sem_dev_raw, color="Semantic Raw (Dev)")) +
  geom_line(aes(x=index, y=sem_dev_loop, color="Semantic Loop (Dev)")) +
  labs(title = "Semantic F1 Score", y = "F1 Score", x = "Epoch")
```



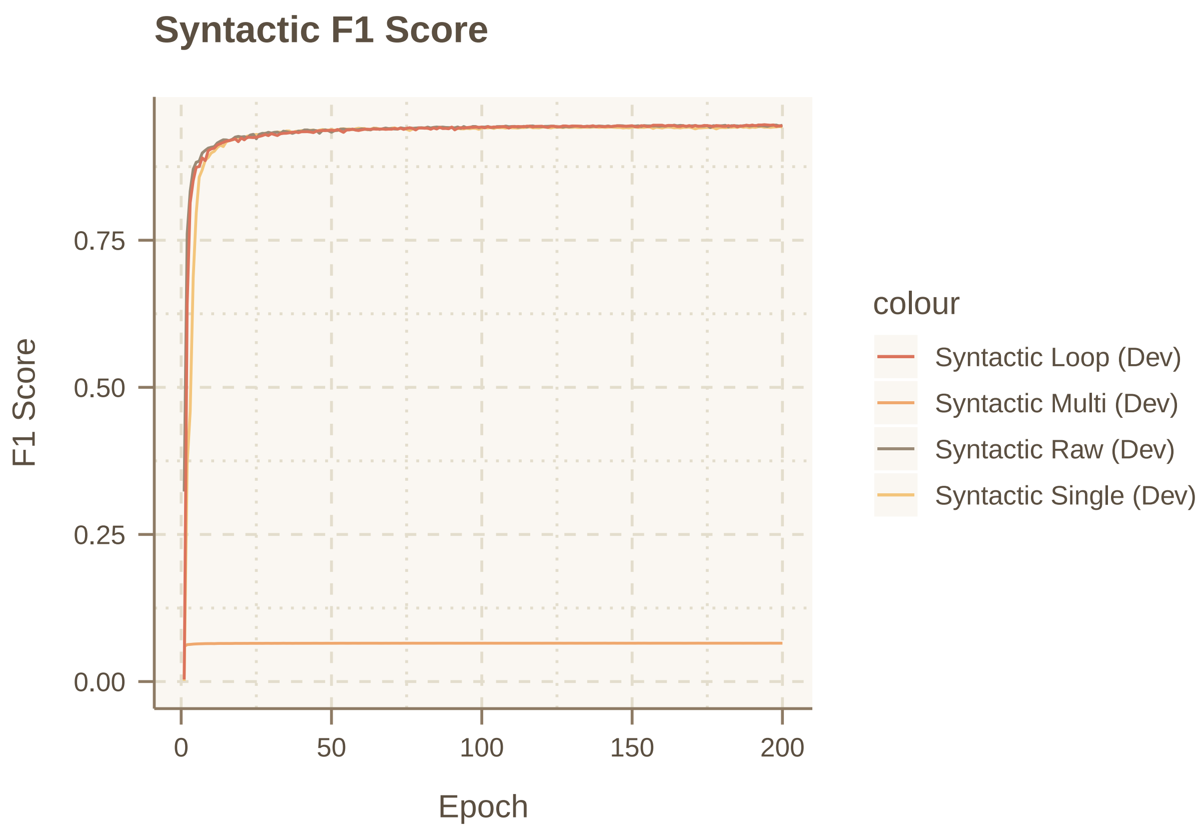
```
# Prints data format for Latex
for (j in 7:10) {
  print(colnames(df)[j])

  for (i in 1:nrow(df)) {
    cat("(", i, ",", " " , df[i,j] , ")\n", sep="")
  }
}
```

```
## [1] "syn_dev_loop"
## (1, 0.003686962)
## (2, 0.6408002)
## (3, 0.8135789)
## (4, 0.8510533)
## (5, 0.8739615)
## (6, 0.8751335)
## (7, 0.8900372)
## (8, 0.8851991)
## (9, 0.9022685)
## (10, 0.9051408)
## (11, 0.9064124)
## (12, 0.9117229)
## (13, 0.9140481)
## (14, 0.9161737)
## (15, 0.9184131)
## (16, 0.9192994)
## (17, 0.9207006)
## (18, 0.9220996)
## (19, 0.9174086)
## (20, 0.9238046)
## (21, 0.9205158)
## (22, 0.9249461)
## (23, 0.9247303)
## (24, 0.9243895)
## (25, 0.9251122)
## (26, 0.9263042)
## (27, 0.9277136)
## (28, 0.9296346)
## (29, 0.9277768)
## (30, 0.9311483)
## (31, 0.9294678)
## (32, 0.927883)
## (33, 0.9308121)
## (34, 0.9316413)
## (35, 0.9317755)
## (36, 0.9325439)
## (37, 0.9337137)
## (38, 0.9339629)
## (39, 0.9327994)
## (40, 0.9343192)
## (41, 0.9343591)
## (42, 0.9344464)
## (43, 0.9338432)
## (44, 0.9327956)
## (45, 0.9353831)
## (46, 0.9361107)
## (47, 0.9363642)
## (48, 0.9361945)
## (49, 0.9366357)
## (50, 0.9370674)
## (51, 0.9362817)
## (52, 0.9382213)
## (53, 0.9358574)
## (54, 0.933248)
## (55, 0.937353)
## (56, 0.9374584)
## (57, 0.9386863)
## (58, 0.9371383)
## (59, 0.9364065)
## (60, 0.9375099)
## (61, 0.9384735)
## (62, 0.9387733)
## (63, 0.9383889)
## (64, 0.9389863)
## (65, 0.9386711)
## (66, 0.9389874)
## (67, 0.9389942)
## (68, 0.9383778)
## (69, 0.9387318)
## (70, 0.938597)
## (71, 0.9400723)
## (72, 0.9389696)
## (73, 0.9404764)
## (74, 0.9382762)
## (75, 0.9403721)
## (76, 0.9409371)
## (77, 0.9398467)
## (78, 0.9374057)
## (79, 0.9406896)
## (80, 0.9406634)
## (81, 0.9402645)
## (82, 0.9404556)
## (83, 0.9385832)
## (84, 0.9407667)
## (85, 0.9390257)
## (86, 0.9418662)
## (87, 0.9398085)
## (88, 0.9399929)
## (89, 0.9393692)
## (90, 0.9423727)
## (91, 0.9374648)
## (92, 0.9403817)
## (93, 0.9413177)
## (94, 0.9405662)
## (95, 0.9413839)
## (96, 0.9420218)
## (97, 0.9417915)
## (98, 0.9422324)
## (99, 0.9419528)
## (100, 0.9421382)
## (101, 0.9409945)
## (102, 0.9430933)
## (103, 0.9415607)
## (104, 0.9425485)
## (105, 0.9421789)
## (106, 0.9424954)
## (107, 0.9432407)
## (108, 0.9426811)
## (109, 0.9405343)
## (110, 0.9431205)
## (111, 0.9427128)
## (112, 0.9433113)
## (113, 0.9430587)
## (114, 0.9428434)
## (115, 0.9438916)
## (116, 0.9436982)
## (117, 0.9424566)
## (118, 0.94349)
## (119, 0.9436256)
## (120, 0.9437687)
## (121, 0.9424269)
## (122, 0.9418127)
## (123, 0.9436966)
## (124, 0.9436707)
## (125, 0.9429732)
## (126, 0.9426137)
## (127, 0.9442314)
## (128, 0.9441743)
## (129, 0.9434303)
## (130, 0.9442115)
## (131, 0.9440789)
## (132, 0.9438439)
## (133, 0.9439079)
## (134, 0.9427993)
## (135, 0.9435416)
## (136, 0.9436698)
## (137, 0.9445478)
## (138, 0.9427767)
## (139, 0.9437314)
## (140, 0.9429876)
## (141, 0.9431308)
## (142, 0.9435697)
## (143, 0.9431768)
## (144, 0.9439058)
## (145, 0.9444001)
## (146, 0.9441638)
## (147, 0.9436623)
## (148, 0.9437429)
## (149, 0.9442096)
## (150, 0.9446532)
## (151, 0.9428284)
## (152, 0.9427749)
## (153, 0.9446569)
## (154, 0.9443471)
## (155, 0.9431428)
## (156, 0.943516)
## (157, 0.9455078)
## (158, 0.9454008)
## (159, 0.9454468)
## (160, 0.9454464)
## (161, 0.9440546)
## (162, 0.9451458)
## (163, 0.945039)
## (164, 0.9449787)
## (165, 0.9433318)
## (166, 0.9433544)
## (167, 0.9446181)
## (168, 0.9435941)
## (169, 0.9448721)
## (170, 0.942777)
## (171, 0.9450766)
## (172, 0.9434032)
## (173, 0.9439498)
## (174, 0.9450998)
## (175, 0.9446923)
## (176, 0.9442523)
## (177, 0.9431047)
## (178, 0.9449305)
## (179, 0.9443904)
## (180, 0.9434762)
## (181, 0.9433539)
## (182, 0.9448387)
## (183, 0.9435665)
## (184, 0.9448057)
## (185, 0.9425127)
## (186, 0.9444001)
## (187, 0.9439795)
## (188, 0.945545)
## (189, 0.9447172)
## (190, 0.945788)
## (191, 0.9443718)
## (192, 0.9459048)
## (193, 0.9458044)
## (194, 0.9463492)
## (195, 0.9458523)
## (196, 0.9454424)
## (197, 0.9459413)
## (198, 0.9435001)
## (199, 0.9445251)
## (200, 0.9442059)
## [1] "syn_dev_multi"
## (1, 0.05969791)
## (2, 0.06280651)
## (3, 0.06315996)
## (4, 0.06365302)
## (5, 0.06399234)
## (6, 0.06419027)
## (7, 0.06429454)
## (8, 0.06449247)
## (9, 0.06454902)
## (10, 0.06457376)
## (11, 0.06449424)
## (12, 0.06473458)
## (13, 0.06478937)
## (14, 0.06476816)
## (15, 0.06479644)
## (16, 0.06476993)
## (17, 0.06480704)
## (18, 0.06494312)
## (19, 0.06487773)
## (20, 0.06489364)
## (21, 0.06492898)
## (22, 0.06488834)
## (23, 0.06498023)
## (24, 0.06500497)
## (25, 0.06494489)
## (26, 0.06498554)
## (27, 0.06495196)
## (28, 0.06501381)
## (29, 0.06502441)
## (30, 0.06496079)
## (31, 0.06502972)
## (32, 0.06499614)
## (33, 0.06504209)
## (34, 0.06510748)
## (35, 0.06504916)
## (36, 0.0650686)
## (37, 0.06499084)
## (38, 0.06506506)
## (39, 0.06505092)
## (40, 0.06507036)
## (41, 0.0650951)
## (42, 0.0650951)
## (43, 0.0650951)
## (44, 0.06510924)
## (45, 0.06513752)
## (46, 0.0650951)
## (47, 0.06508804)
## (48, 0.06511101)
## (49, 0.06512692)
## (50, 0.06508097)
## (51, 0.06513929)
## (52, 0.06510394)
## (53, 0.06515342)
## (54, 0.06516403)
## (55, 0.06514282)
## (56, 0.06511985)
## (57, 0.06517463)
## (58, 0.06515696)
## (59, 0.06513575)
## (60, 0.0651711)
## (61, 0.0651711)
## (62, 0.06514989)
## (63, 0.06517463)
## (64, 0.06517817)
## (65, 0.0651923)
## (66, 0.06514636)
## (67, 0.06514636)
## (68, 0.0651817)
## (69, 0.06513929)
## (70, 0.06517817)
## (71, 0.06518877)
## (72, 0.06516756)
## (73, 0.06512515)
## (74, 0.06520467)
## (75, 0.06515519)
## (76, 0.0651711)
## (77, 0.06520114)
## (78, 0.06517286)
## (79, 0.06518523)
## (80, 0.06518523)
## (81, 0.06519937)
## (82, 0.0651764)
## (83, 0.06517463)
## (84, 0.0651658)
## (85, 0.06521881)
## (86, 0.06519407)
## (87, 0.06520467)
## (88, 0.06523118)
## (89, 0.06522765)
## (90, 0.06521351)
## (91, 0.0651764)
## (92, 0.0651711)
## (93, 0.0651711)
## (94, 0.06519054)
## (95, 0.06520998)
## (96, 0.0651711)
## (97, 0.06517817)
## (98, 0.0651658)
## (99, 0.06520821)
## (100, 0.06518523)
## (101, 0.06521174)
## (102, 0.06518523)
## (103, 0.06519584)
## (104, 0.0651764)
## (105, 0.06516226)
## (106, 0.065187)
## (107, 0.06519584)
## (108, 0.06516226)
## (109, 0.065187)
## (110, 0.06519407)
## (111, 0.06519584)
## (112, 0.06520821)
## (113, 0.0651923)
## (114, 0.0651817)
## (115, 0.06516226)
## (116, 0.06518347)
## (117, 0.06519584)
## (118, 0.06520644)
## (119, 0.06522058)
## (120, 0.06515873)
## (121, 0.06520644)
## (122, 0.06521528)
## (123, 0.06522942)
## (124, 0.0651923)
## (125, 0.06520467)
## (126, 0.06519937)
## (127, 0.06518523)
## (128, 0.06518523)
## (129, 0.06519937)
## (130, 0.06522588)
## (131, 0.06518877)
## (132, 0.06517463)
## (133, 0.06519584)
## (134, 0.06515519)
## (135, 0.06517817)
## (136, 0.06523118)
## (137, 0.06519407)
## (138, 0.06522765)
## (139, 0.06522942)
## (140, 0.06520291)
## (141, 0.06524179)
## (142, 0.065187)
## (143, 0.06522235)
## (144, 0.06524709)
## (145, 0.06518523)
## (146, 0.06520821)
## (147, 0.06523295)
## (148, 0.0651711)
## (149, 0.06523649)
## (150, 0.06521881)
## (151, 0.065187)
## (152, 0.06525769)
## (153, 0.06521705)
## (154, 0.06524179)
## (155, 0.06522765)
## (156, 0.06519054)
## (157, 0.06521528)
## (158, 0.06519584)
## (159, 0.06520644)
## (160, 0.06522058)
## (161, 0.06520644)
## (162, 0.06522235)
## (163, 0.06522765)
## (164, 0.06523118)
## (165, 0.06522058)
## (166, 0.06519937)
## (167, 0.0651764)
## (168, 0.06516403)
## (169, 0.06523118)
## (170, 0.06522235)
## (171, 0.06521351)
## (172, 0.06526653)
## (173, 0.06521351)
## (174, 0.06520114)
## (175, 0.0651923)
## (176, 0.06520821)
## (177, 0.06519761)
## (178, 0.06523118)
## (179, 0.06522411)
## (180, 0.06523649)
## (181, 0.06521705)
## (182, 0.06521351)
## (183, 0.06516933)
## (184, 0.06523118)
## (185, 0.06522411)
## (186, 0.06522058)
## (187, 0.06516756)
## (188, 0.06521705)
## (189, 0.06519407)
## (190, 0.06520821)
## (191, 0.06518877)
## (192, 0.06521174)
## (193, 0.06522765)
## (194, 0.06521705)
## (195, 0.06520467)
## (196, 0.06524002)
## (197, 0.06521705)
## (198, 0.06522765)
## (199, 0.06521705)
## (200, 0.06519584)
## [1] "syn_dev_raw"
## (1, 0.3230068)
## (2, 0.7609165)
## (3, 0.8331008)
## (4, 0.8707283)
## (5, 0.8825849)
## (6, 0.8842841)
## (7, 0.8982247)
## (8, 0.9026495)
## (9, 0.9065152)
## (10, 0.9079885)
## (11, 0.9090688)
## (12, 0.9151417)
## (13, 0.9181167)
## (14, 0.9207168)
## (15, 0.9204639)
## (16, 0.9192357)
## (17, 0.9214589)
## (18, 0.9252647)
## (19, 0.9264177)
## (20, 0.9254222)
## (21, 0.9261504)
## (22, 0.9240501)
## (23, 0.9288192)
## (24, 0.9299665)
## (25, 0.9223547)
## (26, 0.9299539)
## (27, 0.9316206)
## (28, 0.9314259)
## (29, 0.9333902)
## (30, 0.9320942)
## (31, 0.9333494)
## (32, 0.9336175)
## (33, 0.9313712)
## (34, 0.9352824)
## (35, 0.9343727)
## (36, 0.9333975)
## (37, 0.9316013)
## (38, 0.9333929)
## (39, 0.9354431)
## (40, 0.9347341)
## (41, 0.9374277)
## (42, 0.9374514)
## (43, 0.9361111)
## (44, 0.9369155)
## (45, 0.936144)
## (46, 0.9316803)
## (47, 0.9367498)
## (48, 0.9375894)
## (49, 0.9359624)
## (50, 0.9335361)
## (51, 0.9361294)
## (52, 0.9367689)
## (53, 0.9386448)
## (54, 0.9389351)
## (55, 0.9381299)
## (56, 0.9379499)
## (57, 0.9386863)
## (58, 0.9377989)
## (59, 0.9379186)
## (60, 0.9391658)
## (61, 0.9386979)
## (62, 0.9381559)
## (63, 0.9374851)
## (64, 0.9397235)
## (65, 0.9393927)
## (66, 0.9383137)
## (67, 0.9394213)
## (68, 0.9406354)
## (69, 0.9396533)
## (70, 0.9397158)
## (71, 0.9389798)
## (72, 0.9387475)
## (73, 0.941033)
## (74, 0.9401301)
## (75, 0.9398004)
## (76, 0.9411012)
## (77, 0.9401807)
## (78, 0.9406028)
## (79, 0.941064)
## (80, 0.9412405)
## (81, 0.9405421)
## (82, 0.942222)
## (83, 0.9407223)
## (84, 0.9417703)
## (85, 0.9423919)
## (86, 0.9418807)
## (87, 0.9423082)
## (88, 0.9415949)
## (89, 0.9414312)
## (90, 0.9411375)
## (91, 0.9409734)
## (92, 0.9425053)
## (93, 0.9401306)
## (94, 0.9431721)
## (95, 0.9409572)
## (96, 0.9414133)
## (97, 0.9432929)
## (98, 0.9432123)
## (99, 0.9410894)
## (100, 0.9424692)
## (101, 0.9422397)
## (102, 0.9432306)
## (103, 0.9418751)
## (104, 0.9415529)
## (105, 0.9430931)
## (106, 0.9431935)
## (107, 0.9429192)
## (108, 0.9439477)
## (109, 0.9428926)
## (110, 0.9433026)
## (111, 0.9434047)
## (112, 0.9421591)
## (113, 0.9426677)
## (114, 0.9435623)
## (115, 0.9424564)
## (116, 0.9440087)
## (117, 0.9442349)
## (118, 0.9430501)
## (119, 0.9430875)
## (120, 0.9429104)
## (121, 0.943511)
## (122, 0.9435695)
## (123, 0.9438897)
## (124, 0.9438193)
## (125, 0.9423705)
## (126, 0.9429278)
## (127, 0.9424463)
## (128, 0.9432585)
## (129, 0.9425992)
## (130, 0.9432446)
## (131, 0.943905)
## (132, 0.9437715)
## (133, 0.9430324)
## (134, 0.9432096)
## (135, 0.9437264)
## (136, 0.9428318)
## (137, 0.9436373)
## (138, 0.9435666)
## (139, 0.9440473)
## (140, 0.9437964)
## (141, 0.9433209)
## (142, 0.9442421)
## (143, 0.9431979)
## (144, 0.9435925)
## (145, 0.9445042)
## (146, 0.9446624)
## (147, 0.9444156)
## (148, 0.9434434)
## (149, 0.9436783)
## (150, 0.9440999)
## (151, 0.9438948)
## (152, 0.9444902)
## (153, 0.9432292)
## (154, 0.9447657)
## (155, 0.9443848)
## (156, 0.9443089)
## (157, 0.9441095)
## (158, 0.9442299)
## (159, 0.9436133)
## (160, 0.9442479)
## (161, 0.943794)
## (162, 0.9449021)
## (163, 0.944381)
## (164, 0.9455496)
## (165, 0.9444157)
## (166, 0.9451544)
## (167, 0.9447564)
## (168, 0.9432056)
## (169, 0.9443302)
## (170, 0.9443921)
## (171, 0.9448369)
## (172, 0.9439355)
## (173, 0.9438065)
## (174, 0.9439992)
## (175, 0.9449392)
## (176, 0.9415353)
## (177, 0.9447061)
## (178, 0.9441102)
## (179, 0.9443737)
## (180, 0.9445791)
## (181, 0.9452846)
## (182, 0.9424979)
## (183, 0.9446314)
## (184, 0.9445404)
## (185, 0.9442723)
## (186, 0.9437212)
## (187, 0.9446295)
## (188, 0.9451457)
## (189, 0.9438995)
## (190, 0.9445484)
## (191, 0.9448821)
## (192, 0.9441548)
## (193, 0.9446408)
## (194, 0.944216)
## (195, 0.9434761)
## (196, 0.9447809)
## (197, 0.9454608)
## (198, 0.9456607)
## (199, 0.9439989)
## (200, 0.9454786)
## [1] "syn_dev_single"
## (1, 0.001264955)
## (2, 0.3697442)
## (3, 0.4590058)
## (4, 0.6865784)
## (5, 0.7958889)
## (6, 0.8571235)
## (7, 0.8696011)
## (8, 0.8859492)
## (9, 0.8906128)
## (10, 0.8983177)
## (11, 0.9011114)
## (12, 0.9074245)
## (13, 0.9116516)
## (14, 0.9091524)
## (15, 0.9171483)
## (16, 0.9191324)
## (17, 0.9205271)
## (18, 0.922012)
## (19, 0.9246582)
## (20, 0.9229191)
## (21, 0.9242384)
## (22, 0.9268403)
## (23, 0.9280898)
## (24, 0.9290929)
## (25, 0.9297277)
## (26, 0.9294554)
## (27, 0.9302165)
## (28, 0.9319242)
## (29, 0.9318492)
## (30, 0.9311305)
## (31, 0.9328385)
## (32, 0.9340034)
## (33, 0.9334738)
## (34, 0.9336073)
## (35, 0.9356929)
## (36, 0.9356447)
## (37, 0.9320488)
## (38, 0.934922)
## (39, 0.935204)
## (40, 0.9362763)
## (41, 0.9363769)
## (42, 0.9361894)
## (43, 0.9369288)
## (44, 0.9372581)
## (45, 0.9357176)
## (46, 0.9369593)
## (47, 0.938087)
## (48, 0.9374284)
## (49, 0.9370842)
## (50, 0.9381126)
## (51, 0.9368995)
## (52, 0.9365427)
## (53, 0.9391733)
## (54, 0.93946)
## (55, 0.9387826)
## (56, 0.9391055)
## (57, 0.9377781)
## (58, 0.9393252)
## (59, 0.9403379)
## (60, 0.9398374)
## (61, 0.9399375)
## (62, 0.9375686)
## (63, 0.9385475)
## (64, 0.9405821)
## (65, 0.940282)
## (66, 0.9390954)
## (67, 0.9390788)
## (68, 0.9393791)
## (69, 0.9394276)
## (70, 0.9404137)
## (71, 0.9409674)
## (72, 0.9392024)
## (73, 0.9400516)
## (74, 0.9394887)
## (75, 0.9384406)
## (76, 0.9364863)
## (77, 0.9389785)
## (78, 0.9404446)
## (79, 0.9408245)
## (80, 0.9404851)
## (81, 0.941046)
## (82, 0.9398548)
## (83, 0.9407479)
## (84, 0.9412485)
## (85, 0.9405161)
## (86, 0.9411796)
## (87, 0.9409424)
## (88, 0.9415204)
## (89, 0.940891)
## (90, 0.9417482)
## (91, 0.9406911)
## (92, 0.9398357)
## (93, 0.9392878)
## (94, 0.9392131)
## (95, 0.9401664)
## (96, 0.9394741)
## (97, 0.9403684)
## (98, 0.9401096)
## (99, 0.9384269)
## (100, 0.9407191)
## (101, 0.9410963)
## (102, 0.9408445)
## (103, 0.9414298)
## (104, 0.9398913)
## (105, 0.9414836)
## (106, 0.9411576)
## (107, 0.9412817)
## (108, 0.9404914)
## (109, 0.9415035)
## (110, 0.9409792)
## (111, 0.9409358)
## (112, 0.9404636)
## (113, 0.9413738)
## (114, 0.9414351)
## (115, 0.9418903)
## (116, 0.9423591)
## (117, 0.9410394)
## (118, 0.9412093)
## (119, 0.9408826)
## (120, 0.9420323)
## (121, 0.9422587)
## (122, 0.941434)
## (123, 0.9408311)
## (124, 0.9419358)
## (125, 0.9417667)
## (126, 0.9420253)
## (127, 0.9413089)
## (128, 0.9421718)
## (129, 0.9417197)
## (130, 0.9417437)
## (131, 0.9420592)
## (132, 0.9415808)
## (133, 0.9416148)
## (134, 0.9421455)
## (135, 0.9418513)
## (136, 0.9418498)
## (137, 0.9421971)
## (138, 0.942375)
## (139, 0.9417405)
## (140, 0.9422711)
## (141, 0.9416814)
## (142, 0.9420161)
## (143, 0.9423498)
## (144, 0.9418713)
## (145, 0.9415753)
## (146, 0.941609)
## (147, 0.9407213)
## (148, 0.941267)
## (149, 0.9408412)
## (150, 0.9421494)
## (151, 0.9423193)
## (152, 0.941563)
## (153, 0.9417557)
## (154, 0.9416027)
## (155, 0.9424666)
## (156, 0.9421868)
## (157, 0.9399926)
## (158, 0.9419307)
## (159, 0.9414074)
## (160, 0.9406706)
## (161, 0.9419132)
## (162, 0.9422249)
## (163, 0.9419034)
## (164, 0.940913)
## (165, 0.9412344)
## (166, 0.9407454)
## (167, 0.9416751)
## (168, 0.941751)
## (169, 0.9420653)
## (170, 0.9409725)
## (171, 0.9392585)
## (172, 0.9407093)
## (173, 0.9409013)
## (174, 0.941261)
## (175, 0.9415591)
## (176, 0.9412935)
## (177, 0.9412626)
## (178, 0.9391626)
## (179, 0.941164)
## (180, 0.9415681)
## (181, 0.941189)
## (182, 0.9417063)
## (183, 0.941793)
## (184, 0.9421766)
## (185, 0.9416853)
## (186, 0.9429881)
## (187, 0.9419014)
## (188, 0.9424644)
## (189, 0.9411391)
## (190, 0.9422275)
## (191, 0.9417541)
## (192, 0.9428279)
## (193, 0.942964)
## (194, 0.9418577)
## (195, 0.9421006)
## (196, 0.9416422)
## (197, 0.94226)
## (198, 0.9424634)
## (199, 0.9430838)
## (200, 0.9426675)
```

```r
ggthemr('dust', layout="scientific", spacing=2)
```

```
## Warning: New theme missing the following elements: axis.ticks.length.x,
## axis.ticks.length.x.top, axis.ticks.length.x.bottom, axis.ticks.length.y,
## axis.ticks.length.y.left, axis.ticks.length.y.right
```

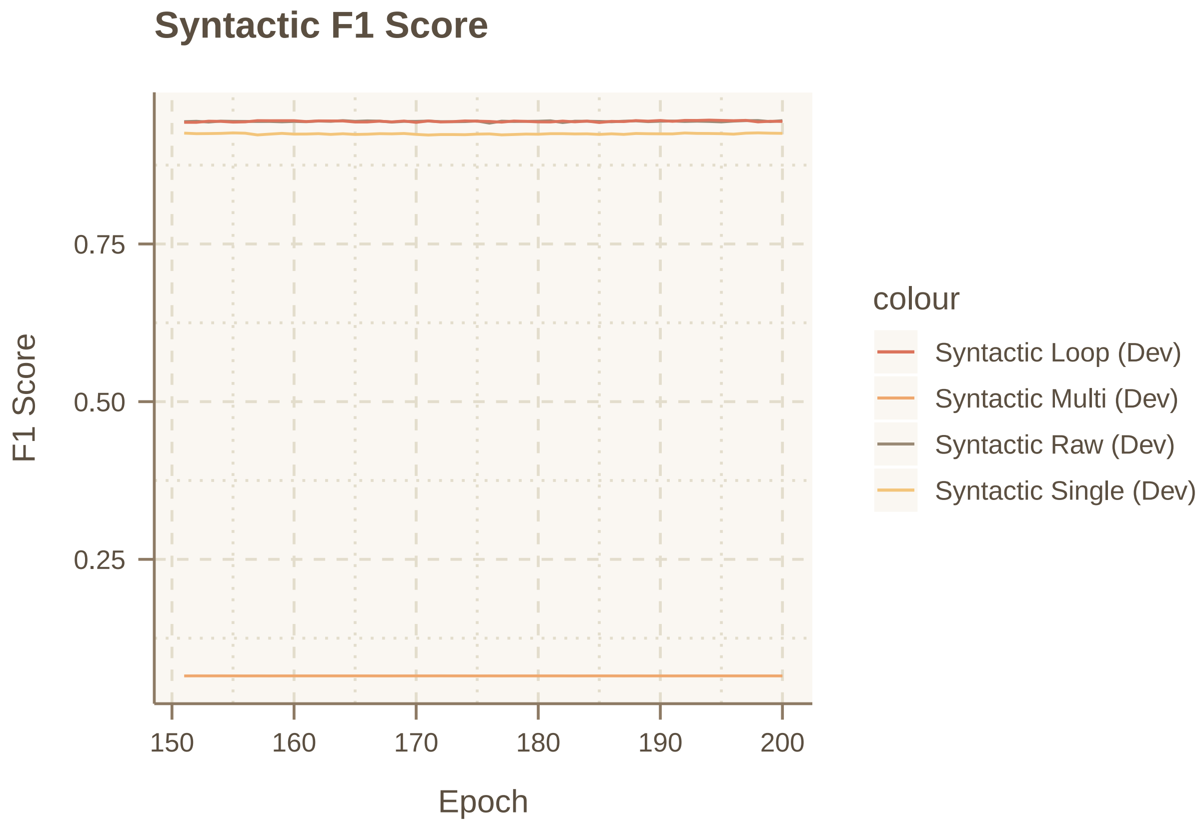
```r
ggplot(df) +
  geom_line(aes(x=index, y=syn_dev_single, color="Syntactic Single (Dev)")) +
  geom_line(aes(x=index, y=syn_dev_multi, color="Syntactic Multi (Dev)")) +
  geom_line(aes(x=index, y=syn_dev_raw, color="Syntactic Raw (Dev)")) +
  geom_line(aes(x=index, y=syn_dev_loop, color="Syntactic Loop (Dev)")) +
  labs(title = "Syntactic F1 Score", y = "F1 Score", x = "Epoch")
```



```
ggthemr('dust', layout="scientific", spacing=2)
```

```
## Warning: New theme missing the following elements: axis.ticks.length.x,
## axis.ticks.length.x.top, axis.ticks.length.x.bottom, axis.ticks.length.y,
## axis.ticks.length.y.left, axis.ticks.length.y.right
```

```
ggplot(df[151:200,]) +
  geom_line(aes(x=index, y=sem_dev_single, color="Syntactic Single (Dev)")) +
  geom_line(aes(x=index, y=syn_dev_multi, color="Syntactic Multi (Dev)")) +
  geom_line(aes(x=index, y=syn_dev_raw, color="Syntactic Raw (Dev)")) +
  geom_line(aes(x=index, y=syn_dev_loop, color="Syntactic Loop (Dev)")) +
  labs(title = "Syntactic F1 Score", y = "F1 Score", x = "Epoch")
```



3.3 One-Way Anova

Will test if the mean of the results is the same (excluding failed syntax training for multitask multi).

```
anova_data_sem = data.frame()
anova_data_syn = data.frame()
df_trimmed_sem = df[151:200, 1:6]
df_trimmed_syn = df[151:200, c(1, 7, 9, 10)] # 8 (syn_dev_multi) gets stuck

for (row in 1:nrow(df_trimmed_sem)) {
  for (col in 2:ncol(df_trimmed_sem)) {
    anova_data_sem = rbind(anova_data_sem,
                       data.frame(value = df_trimmed_sem[row, col],
                                  group = colnames(df_trimmed_sem)[col]))
  }
}

for (row in 1:nrow(df_trimmed_syn)) {
  for (col in 2:ncol(df_trimmed_syn)) {
    anova_data_syn = rbind(anova_data_syn,
                       data.frame(value = df_trimmed_syn[row, col],
                                  group = colnames(df_trimmed_syn)[col]))
  }
}
```

```
# Compute the analysis of variance
res.aov.sem <- aov(value ~ group, data = anova_data_sem)
```

```
# Summary of the analysis
summary(res.aov.sem)
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## group         4 0.0006375 1.594e-04   284.4 <2e-16 ***
## Residuals   245 0.0001373 5.600e-07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Compute the analysis of variance
res.aov.syn <- aov(value ~ group, data = anova_data_syn)
# Summary of the analysis
summary(res.aov.syn)
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## group         2 0.0002521 1.261e-04   171.4 <2e-16 ***
## Residuals   147 0.0001081 7.400e-07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```