

Shallow Discourse Parser Comparison and Combination

Paul Opuchlich, 794745
Malte Klingenberg, 794394

Introduction

The field of text discourse theory has been around for several decades, with first theoretical approaches such as Rhetorical Structure Theory (RST) developed in the 1980s [Mann and Thompson, 1988]. Advancements in machine learning have fueled the development of many different kinds of discourse parsers, but while significant improvements have been made in recent years, the existing systems are still far from perfect. Explicit discourse relations (those marked by a connective such as "and", "because", "after" etc.) can be identified by modern systems with high accuracy, but implicit and entity relations pose a far greater challenge. Because the argument spans of discourse relations can vary wildly (from single clauses to several paragraphs), identifying them is another difficult task.

Because the many available parsers are based on different machine learning algorithms, for example support vector machines (SVMs) and different types of neural networks such as convolutional (CNN) and recurrent (RNN), there might be systematical differences in their performances on different tasks. It is not unreasonable to assume that, for example, one machine learning algorithm might perform better at classifying implicit senses, while another algorithm might be best suited to identifying argument spans.

Our goal in this project was to compare the performances of several different parsers and check whether any systematic patterns can be found. We also tried to find a way to combine several parsers in a way that gives better results than the individual parsers to see whether a "crowd intelligence" approach of combining different systems can be a good way to improve performance.

Dataset and Parsers

For our project, we used data from the 2016 CoNLL shared task [Xue et al., 2016]. We were graciously provided with the output of all submitted parsers for the sense-only challenge, but could not obtain full data including the argument spans. Because only few parsers were publicly available and we were only able to run two of them, we limited our analysis to sense identification.

The methods we used for analysing the sense identification performance can also be used for the argument spans. Below, we present a way to create dummy data for testing purposes. If full data should become available, the analysis can easily be extended to include the argument spans.

The parsers submitted to the CoNLL 2016 shared task differ in their scope. Some limited themselves to a certain subtask, such as identifying explicit and implicit senses.

For our purposes, we included all parsers that completed the sense-only task. A list of them can be found in the evaluation chapter. We will compare the performance of our combination approaches against the highest-scoring parser in the overall task, the oslopots parser [Oepen et al., 2016].

For our combination approaches, we used the output of the parsers on the CoNLL 2016 shared task "test" set as training data and evaluated them and the other parsers on the "blind test" set.

Our Approach

In a first short preprocessing step, we standardised the parser outputs, since some of them differed in minor technical aspects, such as treatment of the "Connective" argument for relations without a connective (implicit relations and EntRels). To generate additional data for testing, we developed a simple randomiser which takes the gold standard relation file and applies some random modifications to it. This may be in the form of changing the identified sense or randomly extending or reducing one or both of the argument spans.

In the next step, the parser outputs are scored according to the CoNLL shared task partial scoring procedure. Since the parsers may miss some relations present in the gold standard and falsely identify relations not found in the gold standard, the gold and parsed relations are aligned to be able to score them accurately. Further details on the aligning step can be found in the CoNLL shared task documentation.

The precision, recall and F1 scores are then calculated for each parser for all relations using the CoNLL shared task methods. We first used these scores to evaluate a possible relationship between parser architecture and performance on different relation types as detailed above.

We then used the F1 scores to implement several methods of combining the parser outputs, which were:

- **best wins:** For each pair of argument spans, check the relation sense identified by each parser and its F1 score for that sense. Choose the sense with the highest F1 score.
- **agreement:** For each pair of argument spans, check the relation sense identified by each parser. Choose the sense which most parsers chose (simple majority voting).
- **probability maximisation:** For each pair of argument spans, check the relation sense identified by each parser and its F1 score for that relation. For each sense, add the F1 scores of all parsers which chose that sense. Chose the sense with the highest aggregated score.
- **best three agreement:** Same as agreement, but only take into account the three identified senses with the highest F1 score.

Our code can be found as commented IPython/Jupyter notebooks at <https://github.com/PaulOpu/sdp-project/tree/master/Coding>.

Evaluation

As a first step, we looked at the inter-parser agreement to get a grasp on the individual and group performance of the parsers. Table 1 shows the inter-parser sense agreement between the three highest scoring parsers "oslopots", "ecnucs" and "steven".

It is rare that all parsers predict the correct sense, with this happening for a third of all occurrences for a contrast relation, but never for restatements. It is far more common that one or two of the parsers get the correct result, with the others predicting a wrong sense. For all relations, this happens in at least half of the cases, with percentages usually being higher than that, for example 200 out of 217 for entity relations and 327 out of 391 for conjunctions.

There are, however, clear differences in difficulty of recognition between the senses. For concessions, results, and restatements, it is more common that all parsers give a wrong result than even a single parser giving a correct one. Apparently, these senses have structures that make them harder to identify for the algorithms. This is consistent with the results we found later, where these senses usually had a lower-than-average or even the lowest F1 score among all senses.

An interesting thing to note is that it sometimes occurs that all three parsers agree on the same wrong sense. This is especially common for condition (7 out of 12) and succession (14 out of 24) relations. Often, these are wrongly classified as entity relations, which might be because the parsers are not sure of a discourse relation but identify the entities in the two arguments. For succession relations, 10 of the 14 were classified as synchrony, so the parsers did select the right sense category, but not the right sense.

| sense | total | all correct | at least one correct | all wrong | same wrong |
|----------------------|-------|-------------|----------------------|-----------|------------|
| Comp.Concession | 33 | 3 | 14 | 19 | 4 |
| Comp.Contrast | 399 | 131 | 296 | 103 | 20 |
| Cont.Cause.Reason | 195 | 27 | 120 | 75 | 10 |
| Cont.Cause.Result | 135 | 15 | 60 | 75 | 5 |
| Cont.Condition | 63 | 30 | 51 | 12 | 7 |
| EntRel | 217 | 58 | 200 | 17 | 2 |
| Exp.Alternative | 5 | 0 | 5 | 0 | 0 |
| Exp.Conjunction | 391 | 103 | 327 | 64 | 13 |
| Exp.Instantiation | 91 | 14 | 56 | 35 | 4 |
| Exp.Restatement | 198 | 0 | 84 | 114 | 9 |
| Temp.Asyn.Precedence | 45 | 19 | 38 | 7 | 0 |
| Temp.Asyn.Succession | 69 | 15 | 45 | 24 | 14 |
| Temp.Synchrony | 76 | 16 | 61 | 15 | 3 |

Table 1: Sense agreement (implicit and explicit) between oslopots, ecnucs and steven

Individual statistics for the three best parsers are shown in table 2, with "total" listing the number of predictions made by the parsers for the given sense. The column labeled "only" has the number of relations for which only that parser (out of the three) was able to correctly identify the relation sense. This is a far more common occurrence for the oslopots (180 total) and ecnucs (166 total) parsers than for the steven parser (56 total). This is important to note for the parser combination approaches, since the oslopots and ecnucs parsers can identify most of the relations sense that the steven parser identifies, while the reverse is not true. The column "not found" has the number of relations that could not be aligned. A reason for this might be that the parser chooses the wrong word or phrase as the connective, but this happens only very rarely.

Table 3 shows the pairwise agreement between the three parsers. The column "one correct" is the number of relations correctly identified by at least one of the two parsers, while "both correct" counts the times the parsers agreed on the right answer. In general,

| sense | oslopots | | | | ecnucs | | | | steven | | | |
|----------------------|----------|---------|------|-----------|--------|---------|------|-----------|--------|---------|------|-----------|
| | total | correct | only | not found | total | correct | only | not found | total | correct | only | not found |
| Comp.Concession | 27 | 12 | 1 | 0 | 25 | 11 | 1 | 0 | 34 | 5 | 1 | 0 |
| Comp.Contrast | 275 | 247 | 22 | 1 | 311 | 239 | 22 | 2 | 308 | 181 | 12 | 0 |
| Cont.Cause.Reason | 165 | 92 | 25 | 1 | 152 | 82 | 22 | 2 | 99 | 42 | 4 | 6 |
| Cont.Cause.Result | 54 | 42 | 5 | 0 | 70 | 37 | 13 | 0 | 77 | 33 | 5 | 1 |
| Cont.Condition | 49 | 49 | 3 | 3 | 44 | 43 | 0 | 1 | 63 | 37 | 0 | 3 |
| EntRel | 534 | 177 | 42 | 0 | 245 | 120 | 11 | 0 | 308 | 102 | 6 | 0 |
| Exp.Alternative | 5 | 4 | 1 | 0 | 4 | 3 | 0 | 0 | 3 | 2 | 0 | 0 |
| Exp.Conjunction | 392 | 264 | 41 | 1 | 314 | 231 | 23 | 3 | 427 | 178 | 20 | 1 |
| Exp.Instantiation | 42 | 33 | 5 | 0 | 66 | 49 | 21 | 0 | 36 | 17 | 1 | 0 |
| Exp.Restatement | 122 | 40 | 27 | 0 | 144 | 55 | 42 | 0 | 5 | 3 | 1 | 0 |
| Temp.Asyn.Precedence | 35 | 34 | 2 | 2 | 40 | 31 | 1 | 0 | 58 | 26 | 1 | 0 |
| Temp.Asyn.Succession | 42 | 41 | 3 | 3 | 34 | 33 | 1 | 1 | 42 | 25 | 2 | 2 |
| Temp.Synchrony | 84 | 49 | 3 | 2 | 88 | 51 | 9 | 4 | 71 | 25 | 3 | 1 |

Table 2: Statistics for the three individual parsers. "only" means that the parser was the only one (out of the three) to correctly identify the sense.

the agreement between the oslopots and ecnucs parsers is higher than the agreement between the steven parser and either of the two others. This reflects the fact found above that there are only few cases where the steven parser is the only one to get the relation sense right.

This also has implications on the combination approaches, in particular the best-three-agreement method. If two parsers often both correctly identify a sense, adding a third parser to the combination will rarely improve the accuracy, but may actually worsen it by amplifying the wrong decision in a case where the original two parsers did not come up with the same relation sense.

| sense | oslopots/ecnucs | | ecnucs/steven | | oslopots/steven | |
|----------------------|-----------------|--------------|---------------|--------------|-----------------|--------------|
| | one correct | both correct | one correct | both correct | one correct | both correct |
| Comp.Concession | 13 | 10 | 13 | 3 | 13 | 4 |
| Comp.Contrast | 284 | 202 | 274 | 146 | 274 | 154 |
| Cont.Cause.Reason | 116 | 58 | 95 | 29 | 98 | 36 |
| Cont.Cause.Result | 55 | 24 | 55 | 15 | 47 | 28 |
| Cont.Condition | 51 | 41 | 48 | 32 | 51 | 35 |
| EntRel | 194 | 103 | 158 | 64 | 189 | 90 |
| Exp.Alternative | 5 | 2 | 4 | 1 | 5 | 1 |
| Exp.Conjunction | 307 | 188 | 286 | 123 | 304 | 138 |
| Exp.Instantiation | 55 | 27 | 51 | 15 | 35 | 15 |
| Exp.Restatement | 83 | 12 | 57 | 1 | 42 | 1 |
| Temp.Asyn.Precedence | 37 | 28 | 36 | 21 | 37 | 23 |
| Temp.Asyn.Succession | 43 | 31 | 42 | 16 | 44 | 22 |
| Temp.Synchrony | 60 | 40 | 58 | 18 | 52 | 22 |

Table 3: Pairwise agreement between the three parsers. "one correct" includes relations correctly identified by both parsers ("both correct")

Finally, the F1 scores for all parsers and relations senses are shown in table 4. Also shown is the underlying machine learning concept of each parser as listed in [Xue et al., 2016]. We can again identify that some senses seem to be more difficult than others as they have low F1 scores for all parsers. Concessions, results, and restatements again appear to be the most difficult senses to identify, while, for example, contrasts and conjunctions appeared to be more easy to classify.

Because the architectures of the parsers vary greatly, we were not able to form meaningful categories for performance comparison as the same machine learning method may be applied in different ways. Moreover, some of the parsers used additional support materials and may therefore not be directly comparable to other parsers using the same machine learning method. Also, not all teams that submitted a parser to the CoNLL shared task also submitted a system paper describing their approach.

Combination approaches The results of our combination approaches are shown in tables 5 and 6. The "best wins" approach gave the worst results, with the overall F1 score being 11% lower than the oslopots results. This might be because selecting only one parser can skip over the good parsers and select one that gave good results in training, but performs poorly on the test data. The other results are all close to the oslopots score. Probability maximisation and best three agreement both give slightly worse results than the oslopots parser, probably due to the same problems as mentioned above. This might also be because, as mentioned above, if two parsers already agree in many cases, adding further parsers may actually decrease the overall accuracy.

The agreement method however, where a simple majority voting between the parsers is utilised, improves the overall F1 score by about 1%. While the F1 scores for individual relation senses might decrease slightly (not more than 1%), for some senses it increases by several percent, for example for Temporal.Asynchronous.Precedence (0.8261 \rightarrow 0.8571).

While the improvements made by combining the parsers are not huge, they are significant and can be used if further slight improvement of the results is desired.

| sense | steven - | oslopots SVM | ecnucs linear, CNN | tao0920 SVM, CNN | goethe SVM, FF-NN | nguyenlab random forest | clac CRF, CNN | PurdueNLP SVM, FF-NN | gw0 RNN | ykido SVM | gtalp - |
|----------------------|-------------|-----------------|-----------------------|---------------------|----------------------|----------------------------|------------------|-------------------------|------------|--------------|------------|
| Comp.Concession | 0.0746 | 0.2000 | 0.1897 | 0.2258 | 0.1538 | 0.1571 | 0.1400 | 0.2131 | 0 | 0.3784 | 0.4070 |
| Comp.Contrast | 0.2560 | 0.3665 | 0.3366 | 0.3467 | 0.3250 | 0.3580 | 0.3292 | 0.3375 | 0.0073 | 0.3468 | 0.3526 |
| Cont.Cause.Reason | 0.1429 | 0.2556 | 0.2363 | 0.2522 | 0.2332 | 0.2048 | 0.2088 | 0.2371 | 0.0423 | 0.1842 | 0.2538 |
| Cont.Cause.Result | 0.1557 | 0.2222 | 0.1805 | 0.1946 | 0.2258 | 0.2064 | 0.1803 | 0.1717 | 0 | 0.2241 | 0.2136 |
| Cont.Condition | 0.2937 | 0.4375 | 0.4019 | 0.4112 | 0.3810 | 0.4128 | 0.3700 | 0.3832 | 0 | 0.4273 | 0.4370 |
| EntRel | 0.1943 | 0.2357 | 0.2597 | 0.2700 | 0.2347 | 0.2458 | 0.2008 | 0.2611 | 0 | 0.1733 | 0.2546 |
| Exp.Alternative | 0.2500 | 0.4000 | 0.3333 | 0.3846 | 0.4444 | 0.4545 | 0.4545 | 0.4000 | 0 | 0.3333 | 0.4545 |
| Exp.Conjunction | 0.2176 | 0.3372 | 0.3277 | 0.3248 | 0.3239 | 0.3203 | 0.2914 | 0.3051 | 0.1168 | 0.3259 | 0.3241 |
| Exp.Instantiation | 0.1339 | 0.2481 | 0.3121 | 0.3099 | 0.2484 | 0.1890 | 0.1574 | 0.2432 | 0 | 0.1574 | 0.2785 |
| Exp.Restatement | 0.0148 | 0.1250 | 0.1608 | 0.1278 | 0.1470 | 0.0906 | 0.0889 | 0.1085 | 0.1440 | 0.0234 | 0.1037 |
| Temp.Asyn.Precedence | 0.2524 | 0.4250 | 0.3647 | 0.3827 | 0.3418 | 0.4198 | 0.3816 | 0.3684 | 0 | 0.3784 | 0.4070 |
| Temp.Asyn.Succession | 0.2252 | 0.3694 | 0.3204 | 0.3694 | 0.3426 | 0.3534 | 0.2887 | 0.3113 | 0 | 0.3030 | 0.3030 |
| Temp.Synchrony | 0.1701 | 0.3063 | 0.3110 | 0.3000 | 0.3082 | 0.3164 | 0.2988 | 0.3195 | 0.0721 | 0.3193 | 0.3000 |

Table 4: F1 scores for all parsers and relation senses. Parser architectures were taken from [Xue et al., 2016]. The low values for gw0 result from the fact that that parser had a smaller sense set.

| sense | oslopots | | | best wins | | | agreement | | |
|----------------------|----------|--------|--------|-----------|--------|--------|-----------|--------|--------|
| | P | R | F | P | R | F | P | R | F |
| *Micro-Average | 0.5485 | 0.5476 | 0.5480 | 0.4334 | 0.4334 | 0.4334 | 0.5559 | 0.5550 | 0.5555 |
| Comp.Concession | 1.0000 | 0.0660 | 0.1239 | 1.0000 | 0.0000 | 0.0000 | 1.0000 | 0.0660 | 0.1239 |
| Comp.Contrast | 0.2160 | 0.4909 | 0.3000 | 0.1360 | 0.5636 | 0.2191 | 0.2500 | 0.4909 | 0.3313 |
| Cont.Cause.Reason | 0.4267 | 0.4384 | 0.4324 | 0.4688 | 0.2027 | 0.2830 | 0.4051 | 0.4384 | 0.4211 |
| Cont.Cause.Result | 0.6000 | 0.3000 | 0.4000 | 0.5000 | 0.0204 | 0.0392 | 0.5926 | 0.3200 | 0.4156 |
| Cont.Condition | 0.8667 | 1.0000 | 0.9286 | 0.7879 | 1.0000 | 0.8814 | 0.9286 | 1.0000 | 0.9630 |
| EntRel | 0.4306 | 0.7600 | 0.5497 | 0.4151 | 0.2200 | 0.2876 | 0.4262 | 0.7800 | 0.5512 |
| Exp.Alternative | 1.0000 | 0.3333 | 0.5000 | 1.0000 | 0.3333 | 0.5000 | 1.0000 | 0.3333 | 0.5000 |
| Exp.Conjunction | 0.6704 | 0.7368 | 0.7021 | 0.4595 | 0.8369 | 0.5932 | 0.6839 | 0.7368 | 0.7094 |
| Exp.Instantiation | 0.6000 | 0.1364 | 0.2222 | 0.4762 | 0.2273 | 0.3077 | 0.5455 | 0.1364 | 0.2182 |
| Exp.Restatement | 0.4923 | 0.2133 | 0.2977 | 0.6667 | 0.0132 | 0.0260 | 0.4789 | 0.2267 | 0.3077 |
| Temp.Asyn.Precedence | 0.9048 | 0.7600 | 0.8261 | 0.5063 | 0.8000 | 0.6202 | 0.9512 | 0.7800 | 0.8571 |
| Temp.Asyn.Succession | 0.9600 | 0.7500 | 0.8421 | 0.9216 | 0.7344 | 0.8174 | 0.9600 | 0.7500 | 0.8421 |
| Temp.Synchrony | 0.5538 | 0.6792 | 0.6102 | 0.5439 | 0.6200 | 0.5794 | 0.5606 | 0.6981 | 0.6218 |
| Overall | 0.5485 | 0.5476 | 0.5480 | 0.4334 | 0.4334 | 0.4334 | 0.5559 | 0.5550 | 0.5555 |

Table 5: Performance comparison between the oslopots parser and our combination approaches. Senses are abbreviated for space reasons.

| sense | oslopots | | | prob. maximisation | | | best three agreement | | |
|----------------------|----------|--------|--------|--------------------|--------|--------|----------------------|--------|--------|
| | P | R | F | P | R | F | P | R | F |
| *Micro-Average | 0.5485 | 0.5476 | 0.5480 | 0.5360 | 0.5352 | 0.5356 | 0.5423 | 0.5409 | 0.5416 |
| Comp.Concession | 1.0000 | 0.0660 | 0.1239 | 1.0000 | 0.0660 | 0.1239 | 0.8571 | 0.0566 | 0.1062 |
| Comp.Contrast | 0.2160 | 0.4909 | 0.3000 | 0.2347 | 0.4182 | 0.3007 | 0.2090 | 0.5091 | 0.2963 |
| Cont.Cause.Reason | 0.4267 | 0.4384 | 0.4324 | 0.4706 | 0.4384 | 0.4539 | 0.4756 | 0.5342 | 0.5032 |
| Cont.Cause.Result | 0.6000 | 0.3000 | 0.4000 | 0.5455 | 0.2449 | 0.3380 | 0.4571 | 0.3200 | 0.3765 |
| Cont.Condition | 0.8667 | 1.0000 | 0.9286 | 0.9286 | 1.0000 | 0.9630 | 0.8966 | 1.0000 | 0.9455 |
| EntRel | 0.4306 | 0.7600 | 0.5497 | 0.3824 | 0.7150 | 0.4983 | 0.4238 | 0.6400 | 0.5100 |
| Exp.Alternative | 1.0000 | 0.3333 | 0.5000 | 1.0000 | 0.3333 | 0.5000 | 1.0000 | 0.3333 | 0.5000 |
| Exp.Conjunction | 0.6704 | 0.7368 | 0.7021 | 0.6512 | 0.7377 | 0.6918 | 0.6817 | 0.7469 | 0.7128 |
| Exp.Instantiation | 0.6000 | 0.1364 | 0.2222 | 0.3750 | 0.0682 | 0.1154 | 0.6111 | 0.2500 | 0.3548 |
| Exp.Restatement | 0.4923 | 0.2133 | 0.2977 | 0.4512 | 0.2450 | 0.3176 | 0.4267 | 0.2133 | 0.2844 |
| Temp.Asyn.Precedence | 0.9048 | 0.7600 | 0.8261 | 0.9737 | 0.7400 | 0.8409 | 0.6909 | 0.7600 | 0.7238 |
| Temp.Asyn.Succession | 0.9600 | 0.7500 | 0.8421 | 0.9592 | 0.7344 | 0.8319 | 0.9066 | 0.7500 | 0.8421 |
| Temp.Synchrony | 0.5538 | 0.6792 | 0.6102 | 0.5902 | 0.6923 | 0.6372 | 0.5932 | 0.6731 | 0.6306 |
| Overall | 0.5485 | 0.5476 | 0.5480 | 0.5360 | 0.5352 | 0.5356 | 0.5423 | 0.5409 | 0.5416 |

Table 6: Performance comparison between the oslopots parser and our combination approaches. Senses are abbreviated for space reasons.

Conclusion

Because the parser architectures varied greatly, we were not able to identify any systematic relationships between parser architecture or underlying machine learning method and sense classification performance on certain relation senses. We also only had access to the sense-only data and could therefore not extend our analysis to the argument spans predicted by the parsers. We were, however, able to identify similarities across all parsers regarding the overall "difficulty" of certain relation senses, with some senses being more difficult to identify reliably for all parsers than other senses.

Furthermore, we have shown that using simple statistical methods to combine several different parsers can slightly increase the sense classification accuracy. The improvement is not that large because for the "easy" senses most of the parsers already agreed, while there was nothing we could do in cases where all or most of the parsers calculated a wrong sense. Nevertheless, we were able to improve the accuracy by 0.75% compared to the best parser submitted to the CoNLL shared task.

In the end, our possibilities for analysis were limited by the data available to us. Repeating this investigation with more data may give further interesting insights.

References

- [Mann and Thompson, 1988] Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.
- [Oepen et al., 2016] Oepen, S., Read, J., Scheffler, T., Sidarenka, U., Stede, M., Velldal, E., and Øvrelid, L. (2016). Opt: Oslo–potSDam–teesside. pipelining rules, rankers, and classifier ensembles for shallow discourse parsing. *Proceedings of the CoNLL-16 shared task*, pages 20–26.
- [Xue et al., 2016] Xue, N., Ng, H. T., Pradhan, S., Rutherford, A., Webber, B., Wang, C., and Wang, H. (2016). Conll 2016 shared task on multilingual shallow discourse parsing. *Proceedings of the CoNLL-16 shared task*, pages 1–19.