

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 [**CHECK**] 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.

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. **WHAT DO THEY AGREE ON INSTEAD?**

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 evaluation and pairwise agreement

compare performances of all parsers

relationship between architecture and results?

The results of our combination approaches are shown in tables 3 and 4. 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.

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 → 0.8571). **[EXPAND?]**

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.

Conclusion

parsers are(n't) good at different things

combination (does not) improve results

	steven	oslopots	cenucs	tao0920	goethe	nguyenlab	clac	PurdueNLP	gw0	ykido	gtntp
sense	-	SVM	linear, CNN	SVM, CNN	SVM, FF-NN	random forest	CRF, CNN	SVM, FF-NN	RNN	SVM	-
Comp.Concession	0.0746										
Comp.Contrast	0.2560										
Cont.Cause.Reason	0.1429										
Cont.Cause.Result	0.1557										
Cont.Condition	0.2937										
EntRel	0.1943										
Exp.Alternative	0.2500										
Exp.Conjunction	0.2176										
Exp.Instantiation	0.1339										
Exp.Restatement	0.0148										
Temp.Asyn.Precedence	0.2524										
Temp.Asyn.Succession	0.2252										
Temp.Synchrony	0.1701										

Table 2: text

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 3: 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 4: Performance comparison between the oslopots parser and our combination approaches. Senses are abbreviated for space reasons.

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.