Divide and Extract – Disentangling Clause Splitting and Proposition Extraction

Darina Gold **Torsten Zesch**
Language Technology Lab
University of Duisburg-Essen, Germany
{darina.gold,torsten.zesch}@uni-due.de

Abstract

Proposition extraction from sentences is an important task for information extraction systems. Evaluation of such systems usually conflates two aspects: splitting complex sentences into clauses and the extraction of propositions. It is thus difficult to independently determine the quality of the proposition extraction step.

We create a manually annotated proposition dataset from sentences taken from restaurant reviews that distinguishes between clauses that need to be split and those that do not. The resulting proposition evaluation dataset allows us to independently compare the performance of proposition extraction systems on simple and complex clauses.

Although performance drastically drops on more complex sentences, we show that the same systems perform best on both simple and complex clauses. Furthermore, we show that specific kinds of subordinate clauses pose difficulties to most systems.

1 Introduction

Propositions are predicate-centered tuples consisting of the *verb*, the *subject*, and other *arguments* such as *objects* and *modifiers*. For example in Figure 1, "smiled" is the *predicate* and the other elements are *arguments*. The first argument is

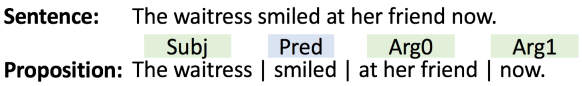

Figure 1: Example Sentence and Extracted Proposition

reserved for the role of the *subject*, in this case "The waitress", while "at her friend" and "now" are arguments, without further sub-specification. Propositions are used in language understanding tasks such as relation extraction (Riedel et al., 2013; Petroni et al., 2015), information retrieval (Löser et al., 2011; Giri et al., 2017), question answering (Khot et al., 2017), word analogy detection (Stanovsky et al., 2015), knowledge base construction (Dong et al., 2014; Stanovsky and Dagan, 2016), summarization (Melli et al., 2006), or other tasks that need comparative operations, such as equality, entailment, or contradiction, on phrases or sentences.

The main goal of this paper is to empirically measure the influence of sentence complexity on the performance of proposition extraction systems. Complexity worsens the extraction of dependencies, on which propositions are built. Hence, proposition extraction performance should decrease with increasing sentence complexity.

The contribution of this work is threefold a) a gold standard corpus for propositions[1], b) an analysis of proposition extraction systems without the influence of complex sentences, and c) an analysis of proposition extraction systems with the influence of complex sentences.

The knowledge of how proposition extraction systems perform on complex sentences will 1) help to identify the system that deals with them best 2) by showing the difficulty with complexity, give a direction towards which proposition extraction systems can be improved.

If different systems perform well on simple or complex sentences, the complexity distinction could help to identify the complexity of a sentence. The complexity of a sentence would then give a direction towards which system would be better to use.

2 Related Work

Proposition are relational tupels extracted from sentences in the form of *predicate-argument struc-*

[1] https://github.com/MeDarina/review_propositions

Proceedings of Recent Advances in Natural Language Processing, pages 399–408,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_047

tures (Marcus et al., 1994). There are proposition models that further distinguish between the type of arguments. They do not only identify the subject, but more complex roles such as temporal and locational objects or causal clauses.

Besides the theory and formalization of proposition, proposition extraction systems have performance issues on real data.

2.1 Comparison of Proposition Systems

Although there have been comparative studies of proposition extraction systems, there has been no extensive study on the impact of sentence complexity on proposition extraction system performance.

Comparative Studies Niklaus et al. (2018) presented an overview of proposition extraction systems and classified them into the classic categories of learning-based, rule-based, and clause-based approaches, as well as approaches capturing inter-propositional relationships. They described the specific problems each system tackles as well as gaps on the overall evolution of proposition extraction systems.

Schneider et al. (2017) present a benchmark for analyzing errors in proposition extraction systems. Their classes are *wrong boundaries*, *redundant extraction*, *wrong extraction*, *uninformative extraction*, *missing extraction*, and *out of scope*. Their pre-defined classes do not map directly to sentence complexity, although *wrong boundaries* and *out of scope* would also be of some interest in an even more detailed error analysis.

Furthermore, according to Stanovsky and Dagan (2016) and Niklaus et al. (2018) there are no common guidelines and followingly no gold standard defining a valid extraction.

Systems Table 1 shows the outputs from different systems, our baselines, and our gold standard.

In their study, Gashteovski et al. (2017) aim at finding a system with minimal attributes, meaning that hedging[2] and attributes expressed e.g. through relative clauses or adjectives, can be optionally removed. Thus, they use recall and two kinds of precision in the evaluation in order to account for the feature of minimality. To explain this in more detail does not lie within the scope of this paper. Gashteovski et al. (2017) evaluates OLLIE (Mausam et al., 2012), ClausIE (Del Corro and Gemulla, 2013), and Stanford OIE (Angeli et al., 2015) against their own system.

Sentence	The waitress smiled at her friend now		
Systems	**Subject**	**Predicate**	**Other Elements**
Allen	The waitress	smiled	at her friend | now
ClausIE	The waitress	smiled	at her friend now
	The waitress	smiled	now
	her	has	friend
ReVerb	The waitress	now smiled at	her friend
Stanford	waitress	smiled at	her friend
	waitress	now smiled at	her friend
OLLIE	The waitress	now smiled at	her friend
OpenIE	The waitress	smiled	now | at her friend
BL1	The	waitress	smiled at her friend now
BL2	The waitress	smiled	at her friend now
Us	The waitress	smiled	at her friend | now

Table 1: Output of Proposition Extraction Systems and Our Two Baselines for the Sentence *The waitress smiled at her friend now*

Stanovsky et al. (2018) evaluates ClausIE, PropS (Stanovsky et al., 2016), and Open IE-4 against their new system, that we will call *Allen* (Stanovsky et al., 2018) herein, using precision-recall, area under the curve, and F1-score. They compare the individual proposition elements. For a proposition to be judged as correct, the predicate and the syntactic heads of the arguments need to be the same as the gold standard.

Saha et al. (2018) evaluate ClausIE, OpenIE-4, and CALMIE (a part of OpenIE) using precision. With the findings of this comparison, they introduce a new version of their system, OpenIE-5[3],

In all described comparisons, the system of the respective authors is the best, which makes sense as it addresses the issue shown by the authors.

2.2 Propositions from Simple Sentences

According to Saha et al. (2018) conjunctive sentences are one of the issues in proposition extraction, as conjunctions are a challenge to dependency parsers (Ficler and Goldberg, 2016) which proposition extraction systems are mostly built upon. Hence, Saha et al. (2018) built a system that automatically creates simple sentences from sentences with several conjunctions that are used for proposition extraction. For the proposition extraction of the simple sentences they used ClausIE and OpenIE. They evaluated their data using three different proposition datasets. The correctness of the extracted proposition from the original sentence were evaluated manually. In their study, simple sentences were sentences without conjunctions.

[2]In pragmatics, hedging is a textual construction that lessens the impact of an utterance. It is often expressed through modal verbs, adjectives, or adverbs.

[3]http://knowitall.github.io/openie/

Quirk (1985) defines a *simple sentence* as a sentence consisting of exactly one independent clause that does not contain any further clause as one of its elements. Hence, a *complex sentence* consists of more than one clause. This is also the definition that we use in our study.

2.3 Crowdsourcing Gold Standard Propositions

Recent work used crowdsourcing for creating and evaluating proposition extraction (Michael et al., 2018; FitzGerald et al., 2018) in the setting of question answering. In short, they asked their crowdworkers to produce questions and answers in a way that resulted in the extraction of their predicates and arguments, without directly asking for predicate-argument structures.

3 Corpus Creation

We create a corpus to evaluate the performance of proposition extraction systems entangled with and disentangled from the task of clause splitting.

Our source corpus is the portion of the Aspect Based Sentiment Analysis (ABSA) task (Pontiki et al., 2014) concerned with restaurant reviews within one aspect – *service*. We use all 423 sentences that were annotated with this aspect. In a preliminary step, we produce a corpus of *reduced* sentences. To examine the influence of sentence complexity, we classify the reduced sentences as either 1) *simple* sentences, meaning sentences with potentially just one proposition, and 2) *complex* sentences, meaning sentences with potentially multiple propositions. Then, we produce propositions from the reduced sentences using expert annotation and evaluate it by calculating the inter-annotator agreement.

Our corpus contains 2,181 sentences (class distribution in Table 2) and 2,526 propositions.

3.1 Preliminary Step: Creating Reduced Sentences

As a preliminary step, we created a gold corpus of reduced sentences formed from originally more complex sentences.

To do so, we use 423 sentences from review texts[4]. As these are quite difficult for producing propositions, even for humans, we included a preliminary step of creating *reduced sentences*. A *reduced sentence* is a sentence that contains only a portion of the original sentence, e.g. the original sentence "The server was cool and served food and drinks" could be reduced to "The server was cool" or "The server served food". The intention behind this step was to create sentences with one proposition only. Hence, the guidelines contained rules such as decomposing conjunctive sentences or creating independent sentences from relative clauses.[5] We perform this preliminary step via crowdsourcing and evaluate it qualitatively.

Definition of Reduced Sentences We instructed our workers to produce reduced sentences from the original sentence. To prevent nested structures, a reduced sentence was not allowed to be split in further reduced sentences, at least within the output of one worker.[6] Ideally, the crowdworkers could have created sentences that contain exactly one proposition. However, this might even be a difficult task for experts, as there are non-trivial sentence constructions that would need long guidelines to create sentences with exactly one proposition. However, our guidelines insured that sentences were reduced in comparison to the original version, if possible. In this way, we are able to create a sufficiently big set of both simple and more complex sentences, as shown in Table 2.

Crowdsourcing We used Amazon Turk for crowdsourcing our data. Michael et al. (2018) crowdsourced gold data for evaluating propositions. The sentence reduction performed here and also in Saha et al. (2018) is very similar to syntactic sentence simplification as performed by Lee and Don (2017). We paid 0.04 $ per HIT and 0.01 $ for each further reduced sentence. Each sentence was reduced by 3 workers. In this process, 2181 unique reduced sentences, which are all used in the following corpus creation process, were created from 423 original sentences.

Evaluation of Reduced Sentences To measure the quality of the crowdsourced reduced sentences, we chose 100 random reduced sentences together with their original sentence and evalu-

[4]Online users' restaurant reviews are a fruitful domain for proposition extraction, as propositions extracted from reviews would be useful for several user-centered tasks, as they would allow to display only information pieces of interest.

[5]However, this step turned out to be more difficult than expected, as some sentences contained several factors that could be reduced. However, this did not influence our goal of determining the influence of sentence complexity.

[6]The annotation instructions are also available on our Github page.

Complexity Class	# of Occurrences
No Verb	101
Simple	1,648
Complex	432
All	2,181

Table 2: Distribution of Sentence Complexity Classes in Our Reduced Sentence Set

ated their correctness using the following non-exclusive categories: ORIGINALSIMPLE, REDUCED, SIMPLE, GRAMMAR, and INFERENCE (see Table 3b).

In Table 3a, we provide an exemplary sentence for each category, except for ORIGINALSIMPLE, as it means that the original is already a simple sentence, containing only one proposition which cannot be further reduced. 20 sentences in the random sample were categorized as being ORIGINALSIMPLE. However, some workers still tried to reduce some of these sentences – 2 of them were grammatically incorrect (GRAMMAR) and 3 fell into the class INFERENCE. This means that their content was not explicitly mentioned in the original sentence, but was lexically inferred.

There were 66 REDUCED sentences, meaning that the sentences have been successfully reduced. 60 of the REDUCED resulted in SIMPLE sentences, which means that they contained only one proposition after the reduction, and 6 were simpler than the original sentence, but contained more than one proposition.

We believe that the results are usable as is, as the error rate is quite low – only 17 of the reduced sentences in the random sample were incorrect (GRAMMAR and INFERENCE), as many of the GRAMMAR errors stem from the original sentence. Furthermore, we show that our reduction step was necessary to produce enough simple sentences for our experiment, as 80% of the random sample were originally complex.

3.2 Creating Propositions from Simple Sentences

To evaluate the performance of proposition extraction systems, we created a gold standard corpus for propositions from the reduced sentences.

In this paper, we follow the most simple possible annotation, similar to Stanovsky et al. (2018).

We want to extract English propositions with one main verb and all arguments that are linked to it. In our notation, the first position of the proposition is the subject, the second is the predicate and the order of the other elements is irrelevant.[7]

The arguments may also contain further propositions, e.g. here, the sentence "I think their food is great" is split in two propositions – "I | think | their food is great" and "their food | is | great ". This definition is restrictive in that it asks for exactly two propositions in the given example. Additionally, it is not bound to a clearly defined theory (as there is no clearly defined theory on propositions). However, it is the representation that is needed to extract information from reviews, as it would help to reduce redundancies, e.g. by clustering sentences such as "Their food is great" and "I think their food is great". Furthermore, we are not interested in inferred information, e.g. "They | have | food" from the previously discussed sentence. This choice will also be reflected in the performance of systems that do not adhere to our understanding of propositions. However, this does not necessarily cloud the performance comparison of simple and complex sentences, as we will still measure the influence of sentence complexity. Each sentence is processed by two annotators and the disagreements are curated in a subsequent step.

Creation As the creation of propositions is not a trivial task, due to many different cases that need to be explained in the guidelines[8], this task should be performed by people who were trained longer than a crowdsourcing platform allows for. Thus, we produced proposition annotations in a double-annotation process by three graduate students[9]. The disagreements were curated by the first author of the paper. The result of the curation builds the gold standard. The gold standard, all annotations, and the guidelines are available.

3.3 Evaluation of Proposition Creation

To evaluate our dataset, we report inter-annotator agreement as well as agreement with the curator

[7]We are not interested in different types of objects and modifiers, similar to Stanford, OpenIE, and AllenNLP, and thus we do not discuss this information. For a better overview, we asked the annotators to present the other elements in their order of occurrence.

[8]The guidelines include explanations of what predicates, arguments, and nested propositions are. This in itself is not difficult. However, such instructions consume more time and need more training, as simple mistakes are made by untrained annotators. We saw this in a training set for this task, that is not included or discussed here due to space restrictions.

[9]The result is shown in Table 4. A1 annotated the whole set, while A2 and A3 annotated parts.

Original Sentence	The server was cool and served food and drinks.
REDUCED	The server was cool and served food.
SIMPLE	The server was cool.
GRAMMAR	The server was.
INFERENCE	The server is good.

(a) Classification Examples

Sentence Class	#
ORIGINALSIMPLE	20
REDUCED	66
SIMPLE	87
GRAMMAR	5
INFERENCE	12

(b) Distribution of Classification

Table 3: Classification of Reduced Sentences

on both the proposition (see Table 4a) and proposition element level (see Table 4b).

Evaluation Metric In order to see differences in the annotation, we performed inter-annotator agreement using %-agreement (accuracy). We use the same measure for system performance, which enables a direct comparison. Although we are aware that agreement is ignorant of chance agreement, we believe that it is the best measure for this problem, as chance agreement is quite low in the case of this complex annotation problem. Furthermore, it is difficult to interpret these results in comparison to other works. As previously described, there are no clear guidelines for propositions and also no manual gold datasets created explicitly for this purpose. We could compare the results of our inter-annotator agreement to similar tasks, where sentences are split into components, as e.g. answers prepared for question answering, paraphrase alignment, translation alignment etc. However, they also have different setups and evaluation metrics and it is out of the scope of this work to discuss these differences.

Levels of Evaluation We perform the evaluation on two levels - *proposition level* and *proposition element level*. On the proposition level, we calculate the agreement of whole propositions. On the proposition element level we calculate the agreement of individual elements of the propositions whilst taking their label (subject, predicate, or other element) into account.

Inter-annotator Agreement Table 4a shows that the inter-annotator agreement on the proposition level is .39 and .53 on complex sentences and .61 and .71 on simple sentences. These agreement differences show that clause splitting is also difficult for humans.

Agreement with Curator The agreement with the curator is .05 to .19 higher than the inter-annotator agreement. The agreement on the proposition element level is .67 and .7 on complex sentences and .83 and .85 for simple sentences - nearly double of the whole proposition agreement.

	Simple		**Complex**		**All**	
	A1	Gold	A1	Gold	A1	Gold
A1	-	.80	-	.66	-	.76
A2	.71	.79	.53	.63	.66	.74
A3	.61	.66	.39	.48	.57	.62

(a) Inter-Annotator Agreement on Propositions

	Simple		**Complex**		**All**	
	A1	Gold	A1	Gold	A1	Gold
A1	-	.90	-	.77	-	.86
A2	.85	.79	.70	.63	.81	.74
A3	.83	.83	.67	.70	.80	.80

(b) Inter-Annotator Agreement on Proposition Elements

Table 4: Inter-Annotator Agreement in Accuracy

4 Evaluation of Proposition Extraction Systems

Similar to Saha et al. (2018); Schneider et al. (2017) and Niklaus et al. (2018), we evaluate proposition system performance. They do not, however, regard the task of proposition extraction disentangled from the intrinsic subtask of clause splitting. By showing the performance of both simple and complex sentences, we are furthermore able to show the impact of clause splitting.

4.1 Setup

To identify the system that performs best when disentangled from the task of clause splitting, we use the herein produced corpus to analyze and evaluate the performance of various proposition extraction systems as used in evaluations by Stanovsky and Dagan (2016), Gashteovski et al. (2017), Saha et al. (2018), and Stanovsky et al.

(2018). Hence, we will analyze proposition extraction performance using AllenNLP, ClausIE, ReVerb, Stanford Open Information Extraction, OLLIE, and OpenIE-5.[10] Furthermore, we provide two baseline systems.

We use %-agreement to measure the performance of systems. We want full agreement, not just matching phrase heads, as performed by Stanovsky et al. (2018). Furthermore, we evaluate only agreement, as in our setup the argument or the predicate matching is what we are interested in, meaning we do not need precision and recall in our setting. In this way, our evaluation setup is similar to Saha et al. (2018), who also identified specific issues in proposition extraction systems.

As in inter-annotator agreement, we calculate agreement on two levels: proposition and proposition element level. The results of the performance comparison is shown in Table 5.

4.2 Baselines

We provide two baselines in order to better compare the systems. Both baselines create propositions with three elements at most: subject, predicate, and one other element. The first baseline (BL1) takes the first word as subject, the second word as predicate and the rest as one other element. The second baseline (BL2) is a little more engineered and uses POS-tags. It makes a proposition for each verb. All words before the verb are the subject and all words after the verb are one other element. Examples for the baselines are shown in the Table 1. The baselines are kept simple on purpose to show how simple algorithms can solve the given problem. A baseline that appears intuitive is using a dependency parser and filtering for the root and its dependants. However, deciding which parts are its dependents and especially the span of arguments is ambiguous. This would not be a baseline, it would be a rule-based system that is not out of the box. Hence, we decided not to do it.

4.3 System Performance

Table 5a shows that performance of proposition extraction on whole propositions is equally bad for both simple and complex sentences. Table 5b shows that performance on proposition elements is much better than on proposition level. Furthermore, the table shows that for all systems but ReVerb, the performance is much better on the simple sentences, which was expected.

It is also interesting that although the performance of both baselines on whole propositions is 0, the performance of the second baseline on proposition elements is competitive. This shows, that the task of proposition extraction can, to a big part, be solved by correct verb extraction. It outperforms ReVerb, Stanford, and on simple and complex sentences also OLLIE. The second baseline performs a little worse on all sentences, as these also include sentences without a verb and this baseline is verb-based. This shows that either the automatic systems have problems with the extraction of verbs or they have deeper issues, e.g. they do not extract from a lot of sentences, as is discussed in Section 4.4.1. The second baseline performs almost equally on both simple and complex sentences. This may show correct verb extraction alone solves only a particular portion of proposition extraction.

Other systems, especially the two best ones, perform about two times better on the simple sentences but then have a much bigger drop on the complex sentences. This may show that clause splitting has a bigger impact on better or probably more intelligent systems than on more simple systems.

On both levels, OpenIE is the best system, very closely followed by Allen, whereas the other systems are well-beaten.

4.4 Analysis of System Performance

Identifying further problems except clause splitting could improve current proposition extraction systems. On the one hand there are sub-issues in clause splitting. On the other hand, there are issues besides clause splitting.

In the case of ClausIE and ReVerb, many further clauses and also arguments are cut, as these consist of a maximum of three elements, which makes the comparison difficult.

4.4.1 General Issues

We first manually examined some potential issues in the proposition extraction from simple sentences. After the manual analysis of potential issues, we calculated the system performance if the issue would be eliminated. One big issue we found is **missing propositions**, meaning that systems do

[10] We will not use MinIE (Gashteovski et al., 2017), as it is an extension of ClausIE providing additional information such as modality and whether an argument is necessary or unnecessary, which is disregarded in this work.

Systems	Simple	Complex	All
Allen	.08	.09	.08
ClausIE	.06	.09	.07
ReVerb	.02	.02	.02
Stanford	.01	.01	.01
OLLIE	.03	.04	.03
OpenIE	**.09**	**.12**	**.09**
BL1	.00	.00	.00
BL2	.00	.00	.00

(a) System Performance on Propositions

Systems	Simple	Complex	All
Allen	.50	.40	.46
ClausIE	.37	.36	.36
ReVerb	.15	.14	.14
Stanford	.20	.09	.17
OLLIE	.24	.19	.22
OpenIE	**.51**	**.42**	**.47**
BL1	.05	.04	.05
BL2	.26	.24	.21

(b) System Performance on Proposition Elements

Table 5: System Performance Measured in Accuracy

Systems	Missing	Conditional	Temporal
Allen	.08	.13	.19
ClausIE	.06	.11	.13
ReVerb	.03	.00	.03
Stanford	.02	.00	.00
OLLIE	.04	.06	.02
OpenIE	.10	.19	.17

(a) System Performance on Propositions Excluding Specific Issues

Systems	Missing	Conditional	Temporal
Allen	.50	.57	.55
ClausIE	.38	.40	.38
Stanford	.26	.03	.14
ReVerb	.32	.00	.21
OLLIE	.31	.00	.20
OpenIE	.54	.53	.50

(b) System Performance on Proposition Elements Excluding Specific Issues

Table 6: System Performance Excluding Specific Issues

not always extract propositions. Except for the missing propositions, there was no big difference in the system performance with or without the issue. Also, some systems have different models of propositions, which may also affect their performance. On the one hand, there are issues with previous steps, e.g. **negations** or **quantifiers** are ignored. On the other hand, there are issues with formatting, e.g. a different treatment of **prepositions** or conditionals.

Missing Propositions One big issue is that proposition extraction systems often do not produce any extraction from a sentence. Unsurprisingly, this issue is bigger among the systems that do not perform well - namely ReVerb (58% of sentences do not have an extraction), Stanford (39%), and OLLIE (33%), whereas the better performing systems have much lower rates - Allen (3%), ClausIE (4%), and OpenIE (10%). In ReVerb, Stanford, and OLLIE we could not find a clear reason why there are no extractions. In the case of Allen, there are only no extractions from sentences without verbs.[11] ClausIE and OpenIE have no extractions from sentences that are missing a verb or a subject. Additionally, OpenIE has no extractions from existential clauses.

In Table 6a, where we show the performances of systems on full propositions without the discussed issues, it is shown that systems perform slightly better when eliminating missing propositions from simple sentences. However, the improvement is clearer in Table 6b on the element level. Especially for the systems that had more missing propositions, namely Stanford, ReVerb, and OLLIE, the change is between .06 - .17.

Conjunctions As already stated by Saha et al. (2018), conjunctive sentences pose an issue to proposition extraction systems. In our case, we wanted to separate all conjunctive sentences in individual propositions, e.g. the sentence "The waitress smiled at her friend and at me." contains the propositions "The waitress | smiled | at her friend" and "The waitress | smiled | at me.". OpenIE and Stanford have the same guidelines on conjunctions, whereas Allen, ClausIE, and ReVerb keep the conjuncted elements together – from the previous sentence they would create one proposition – "The waitress | smiled | at her friend and me.".

Negations Stanford does not extract from negated

[11] These sentences are classified as neither simple nor complex, but are included in all.

sentences and Allen has problems with negated sentences missing a verb. The rest can deal with negations. These specific problems are difficult to show in numbers, as they are rare – only about 7% of the sentences contained negations.

Prepositions OLLIE, ReVerb, and Stanford place the prepositions with the predicate, whereas all other systems as well as our gold standard place it with the associated argument, as is shown in the example in Table 1. For these cases we would need adjusted evaluations that ignore this difference.

Quantifiers Stanford ignores "every" in propositions.

4.4.2 Issues with Complex Sentences

We looked at issues within complex clauses, namely conditional and temporal clauses.

Conditional Clauses In some cases, Allen, ClausIE, OLLIE, and OpenIE extract the if-clause for the argument, but delete the "if", which leads to disagreements on both full proposition and proposition element level. Comparing the performance on all complex clauses as shown in Table 5a to complex clauses without conditional clauses, as shown in Table 6a, all systems, except for ReVerb and Stanford, clearly perform better. Allen is better by .04 and OpenIE by .05, which shows that they have the biggest issues with conditional clauses. On proposition element level this becomes even clearer. Here, the three better systems, ClausIE, Allen, and OpenIE perform .04 - .17 better without conditional clauses.

Temporal Clauses Conceptually, Allen, OLLIE, and OpenIE extract temporal clauses correctly, but have some problems if the sentence is too long. Stanford cuts out the "when". For temporal clauses, the performance is similar to conditional clauses. The three better systems perform .06 -.11 better on full proposition level, and .02-.09 better on proposition element level. Stanford and OLLIE perform worse without the temporal clauses.

5 Summary

In this work, we described a method on how to create a dataset of reduced sentences from originally complex ones. We created an English dataset according to this method and further classified this dataset as simple and complex. It can be used for further evaluation of proposition extraction systems. The dataset enabled us to research the performance of proposition extraction detached from the task of clause splitting.

On the one hand, we showed that sentence complexity has a measurable impact on proposition extraction performance of both humans and machines. Hence, one step towards improving the performance of such systems, is the improvement of clause splitting. Furthermore, we believe that the performance of the original complex sentences, without the preliminary reduction step, would pose an even bigger problem to proposition systems, which implies that using these systems on real data could be problematic.

On the other hand, our study also showed that the ranking of systems is similar among simple and complex sentences. This means, that the best performing systems among simple sentences that are disentangled from the task of clause splitting, are also the best in complex sentences, where clause splitting also needs to be performed. This may mean that to find the overall best system, one does not need to classify between simple and complex sentences. However, it is necessary to find that sentence complexity is one problem of proposition extraction.

Also, our intelligent baseline system, that was able to extract verbs, outperformed three of the systems. However, the better systems did not only perform much better, but they were also more affected by sentence complexity.

Additionally, we looked into further problems of proposition extraction systems. The main issues in complex sentences that we could identify were conditional and temporal clauses.

6 Future Work

In future work, we plan to enlarge the corpus in order to use it for studies on user-specific recommendations. We plan to display proposition-like information to the user to provide more specific information than is given by a long sentence. This work may help in clause splitting, as we not only provide a gold standard for it, but also describe a method on how to create it. Furthermore, we plan to built a proposition extraction system based on the findings from this paper.

Acknowledgement

We would like to thank Venelin Kovatchev and Marie Bexte for their annotations. This work has been funded by Deutsche Forschungsgemeinschaft within the project ASSURE.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging Linguistic Structure For Open Domain Information Extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 1, pages 344–354.

Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: Clause-Based Open Information Extraction. In *Proceedings of the 22nd international conference on World Wide Web*, pages 355–366. ACM.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM.

Jessica Ficler and Yoav Goldberg. 2016. A Neural Network for Coordination Boundary Prediction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 23–32.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-Scale QA-SRL Parsing. *arXiv preprint arXiv:1805.05377*.

Kiril Gashteovski, Rainer Gemulla, and Luciano Del Corro. 2017. Minie: minimizing facts in open information extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2630–2640.

Rachayita Giri, Yosha Porwal, Vaibhavi Shukla, Palak Chadha, and Rishabh Kaushal. 2017. Approaches for information retrieval in legal documents. In *Contemporary Computing (IC3), 2017 Tenth International Conference on*, pages 1–6. IEEE.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering Complex Questions Using Open Information Extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 311–316.

John Lee and J Buddhika K Pathirage Don. 2017. Splitting Complex English Sentences. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 50–55.

Alexander Löser, Sebastian Arnold, and Tillmann Fiehn. 2011. The GoOlap Fact Retrieval Framework. In *European Business Intelligence Summer School*, pages 84–97. Springer.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: annotating predicate argument structure. In *Proceedings of the workshop on Human Language Technology*, pages 114–119. Association for Computational Linguistics.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics.

Gabor Melli, Zhongmin Shi, Yang Wang, Yudong Liu, Anoop Sarkar, and Fred Popowich. 2006. Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2006 Summarization Task. In *Proceedings of the 6th Document Understanding Conference (DUC 2006)*.

Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. 2018. Crowdsourcing Question-Answer Meaning Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 2, pages 560–568.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3866–3878.

Fabio Petroni, Luciano Del Corro, and Rainer Gemulla. 2015. CORE: Context-Aware Open Relation Extraction with Factorization Machines. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1763–1773.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2014)*, pages 27–35.

Randolph Quirk. 1985. *A grammar of contemporary English*, 11. impression edition. Longman, London.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 74–84.

Swarnadeep Saha et al. 2018. Open Information Extraction from Conjunctive Sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2288–2299.

Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A Gers, and Alexander Löser. 2017. Analysing errors of open information extraction systems. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 11–18.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a Large Benchmark for Ipen Information Extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2300–2305.

Gabriel Stanovsky, Ido Dagan, et al. 2015. Open IE as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 303–308.

Gabriel Stanovsky, Jessica Ficler, Ido Dagan, and Yoav Goldberg. 2016. Getting More Out Of Syntax with PROPS. *arXiv preprint arXiv:1603.01648*.

Gabriel Stanovsky, Julian Michael, Luke Zettlemoyer, and Ido Dagan. 2018. Supervised open information extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 885–895.