Effect Graph: Effect Relation Extraction for Explanation Generation

Jonathan Kobbe [1] **Ioana Hulpuș** [2] **Heiner Stuckenschmidt** [1]
[1] University of Mannheim, Germany
[2] Utrecht University, Netherlands
{jonathan,heiner.stuckenschmidt}@uni-mannheim.de
i.r.karnstedt-hulpus@uu.nl

Abstract

Argumentation is an important means of communication. For describing especially arguments about consequences, the notion of *effect relations* has been introduced recently. We propose a method to extract effect relations from large text resources and apply it on encyclopedic and argumentative texts. By connecting the extracted relations, we generate a knowledge graph which we call *effect graph*. For evaluating the effect graph, we perform crowd and expert annotations and create a novel dataset. We demonstrate a possible use case of the effect graph by proposing a method for explaining arguments from consequences.

1 Introduction

Argumentation is a challenging task because its goal is to convince an audience. One broadly used type of arguments is the *argument from consequences*, which has been specifically addressed in recent literature (Reisert et al., 2018; Al-Khatib et al., 2020; Kobbe et al., 2020). The premise of an argument from consequences states that if A is brought about, good or bad consequences will plausibly occur, which leads to the conclusion that A should or should not be brought about (Walton et al., 2008). The following statement is such an argument in favor of legal abortions:

Legal abortions protect women.

At the core of an argument from consequences is what Al-Khatib et al. (2020) call effect relation: A typically expresses either a positive or negative effect on an instance B, which we denote by $A \xrightarrow{+} B$ or $A \xrightarrow{-} B$. In the example, the effect relation is *legal abortions* $\xrightarrow{+}$ *women* because of the positive effect expressed by the verb *protect*. Our main motivation is to further back up such premises by generating structured explanations. Table 1 shows some potential explanations.

1	Abortions protect women from the harm caused by giving birth and being pregnant.
2	Abortions prevent long term damage caused by complications during the pregnancy and birth process.
3	Legal Abortions protect the women's right to self-determination.
4	Abortions protect women from the financial burden of raising a child.
5	Abortions can protect girls from becoming mothers too early.

Table 1: Some possible explanations.

First, we note that it is not possible to find the one and only explanation for why legal abortions protect women. As demonstrated, there exist multiple different explanations and, from merely reading the statement, we cannot know which of these explanations the author had in mind. Thus, our goal is not to *reconstruct* the original explanation, but to *propose* meaningful ones.

For automatically generating possible explanations, we propose an approach that is specific for explaining effect relations. Given $A \rightarrow B$, we aim to find an instance C such that $A \rightarrow C \rightarrow B$. Because of the structure of such an explanation, we call it Effect-Effect-Explanation. Of course, this way, we cannot capture all the details in the explanations in table 1. But we can capture some key aspects and describe the explanations in a well-defined way that allows for further processing in downstream tasks. Table 2 shows possible formalized versions of explanations 1 to 4.

Effect-Effect-Explanations are, however, still very limited in their nature. While we cannot fully overcome this limitation, we show that it is possible to expand upon them for instance by incorporating lexical knowledge: Given $A \rightarrow B$, an explanation could also be ($A \rightarrow C$, C *instanceOf / hypernym / synonym* B) or, vice versa, (A *instanceOf / hy-*

Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (@ACL 2023), pages 116–127

1	Abortions $\xrightarrow{-}$ harm $\xrightarrow{-}$ women
2	Abortions $\xrightarrow{-}$ long term damage $\xrightarrow{-}$ women
3	Legal Abortions $\xrightarrow{+}$ right to self-determination $\xrightarrow{+}$ women
4	Abortions $\xrightarrow{-}$ financial burden $\xrightarrow{-}$ women

Table 2: Formalized Effect-Effect-Explanations.

pernym / synonym C, $C \rightarrow B$). Analoguesly, we call these Effect-Lexical-Explanation. An example for explanation 5 in table 1 would be *Abortions* $\xrightarrow{+}$ *girls* $\xrightarrow{\text{hypernym}}$ *women*.

The main challenge for both of the proposed explanation schemes is to get the additional information (i.e., C and its links to A and B). For the lexical relations, we use WordNet (Fellbaum, 2010). For the effect relations, we propose a simple, yet efficient, extraction method which we denote by EREx (***E**ffect **R**elation **Ex**tractor*). We then apply it on large text resources and connect the extracted relations in a graph which we refer to as effect graph[1]. While we build the graph having explanation generation in mind, it might also be of value for other tasks as it contains a widely used type of knowledge.

In the following, we discuss related work (section 2). In section 3, we describe the generation of the effect graph which we evaluate in section 4. Lastly, we showcase our envisioned explanation generation (section 5) and conclude with a discussion (section 6).

2 Related Work

Our method to extract effect relations is most similar to the one proposed by Kobbe et al. (2020). They extract effect relations in order to classify stances of arguments from consequences. Just as ours, their extraction method is purely heuristic and relies on dependency parsing. The main differences we introduced are due to the following reasons: First, the method of Kobbe et al. (2020) relies on sentence-topic pairs to identify the effect relation's subject, instead of sentences only. Second, it requires the effect relation's object to have a sentiment in order to calculate the stance which is not necessary for our task. Because of this and the first reason, the subjects and objects which are derived by detecting patterns in the dependency parse are no longer controlled for by either linking to the topic or a sentiment lexicon, so we pose other restrictions on both of them. Third, it is designed to extract an effect relation whenever possible, thus emphasizing recall, in order to enable the stance detection. In contrast, we want to rather focus on precision.

Al-Khatib et al. (2020) also extract effect relations from argumentative text and, like ourselves, use them to build a knowledge graph. Their graph is then used as background knowledge by Al Khatib et al. (2021) who use it to support neural argument generation, and by Yuan et al. (2021) who try to identify the correct response to an argument among five possible options. However, in terms of methodology, there are only little similarities to our approach. While EREx is completely unsupervised, Al-Khatib et al. (2020) divide the relation extraction task into several subtasks for which they train specific classifiers, with one exception: For identifying the effect relation's subject and object, they use the supervised OpenIE model of Stanovsky et al. (2018).

OpenIE (Open Information Extraction) is the task to extract relationships between entities from text. In contrast to conventional information extraction, in OpenIE, the relationships are not predefined (Etzioni et al., 2008). However, OpenIE can also be applied for relation extraction with domain specific relations by performing *Relation Mapping* (Soderland et al., 2010). While Soderland et al. (2010) propose a supervised approach, in our case, we consider it sufficient to filter and map the relations using an effect lexicon. Similarly to Corro and Gemulla (2013), Angeli et al. (2015), Gashteovski et al. (2017), we base our relation extraction on dependency parsing. In comparison to these works, however, our effect relation extraction approach is much less sophisticated. Evolving around effect verbs specifically, we use only a small set of manually defined patterns, but are still able to gain comparable or even better results when compared to OpenIE with an effect lexicon based relation mapping.

Similar to our effect graph which we build from effect relations, Martinez-Rodriguez et al. (2018) use ClausIE (Corro and Gemulla, 2013) for extracting relations in order to build an OpenIE-based knowledge graph. Before applying OpenIE, they extract entities and link them to existing knowledge graphs. We experiment with both, using only enti-

[1]The resources created for this paper are available at https://github.com/dwslab/Effect-Graph.

ties which we can link to Wikipedia pages, or not requiring any linking. Further, they annotate noun phrases (NPs) and expand the extracted entities to encompass the complete NP. Similarly, in EREx we only consider NPs as entities.

Lastly, we want to mention another type of relations than effect relations, namely *causal relations* (Davidson, 1967). Other than in effect relations, A's effect on B, if they are in a causal relation, is clearly defined as A being the cause for B. Girju and Moldovan (2002), Girju (2003) introduced the task of automatically extracting causal relations from text, and it has been a matter of research since then (Yang et al., 2022).

Also for causal relations, there exists research on using them for building a knowledge graph. Heindorf et al. (2020) bootstrap dependency parse patterns to extract claimed causal relations from text. While their method to start with a small, very accurate seed set of patterns and to extend it consecutively is very appealing, we find it to be rather difficult to apply on our approach: Their patterns involve very concrete words that all trigger causal relations while we chose to keep our patterns general in order to apply to a large set of different effect words. Also like us, Heindorf et al. (2020) do not fact check their extractions, but emphasize that they merely collect claimed causal relations.

3 Effect Graph Generation

Our aim is to generate a graph where the nodes are entities such as *global warming, CO2 emissions, solar panel*. The edges represent the effect relations and indicate either a negative or positive effect from the source to the target node, e.g., (*solar panel*) $\overset{-}{\rightarrow}$ (*CO2 emissions*). We also store the concrete word indicating the effect. In the previous example, this could be for instance *reduce* or *prevent*.

3.1 Effect Relation Extraction

We use a subset of the dependency parse patterns presented in Kobbe et al. (2020) in order to identify subject and object relations as well as negations. The patterns are presented in table 3.

Using these patterns, we look for triples (S, P, O) such that the predicate P has subject S and object O. In order for the triple to qualify as effect relation, P has to express a positive or negative effect on its object. We identify such effects by applying the Connotation Frame lexicon (Rashkin et al., 2016) with a threshold of ± 0.2, expanded using WordNet as proposed in Kobbe et al. (2020). The effect relation's subject, which we denote by A, is then the statement's substring which is represented by the dependency parse's subtree whose root is S. Analoguesly, the object B is the statement's substring represented by the subtree whose root is O. Thereby, leading articles are ignored and A and B have to be non-stopwords and NPs. To ensure that they are meaningful entities in different contexts, we check whether A and B link to an entry in Wikipedia. Only if they both do, and if neither A nor B nor P are negated, we consider $A \overset{\text{P}}{\rightarrow} B$ to be an effect relation.

	Pattern	*Interpretation*
1	$P \overset{*}{\rightarrow} O$	P has object O
3	$P \overset{\diamond}{\rightarrow} S$	P has subject S
5	$NegP \xrightarrow{pobj} X$	X is negated
6	$X \rightarrow NegP \wedge \nexists NegP \xrightarrow{pobj}$	X is negated
7	$X \xrightarrow{neg}$	X is negated

$* \in \{dobj, cobj, nsubjpass, csubjpass\}$;
$\diamond \in \{nsubj, csubj\}$;
$NegP$ stands for *negative preposition*

Table 3: Dependency graph patterns, adapted from Kobbe et al. (2020).

3.2 Graph Construction

For building the effect graph, we extract effect relations from the following three datasets:

Debatepedia Debatepedia was an online portal where users could add pro and contra arguments to a variety of topics. We use the *featured debates* which overall have high quality.

Debate.org As Debatepedia is rather small, we also use Debate.org (Durmus and Cardie, 2018, 2019) to extract effect relations from a large argumentative text basis. In Debate.org, two users engage in a debate about a certain topic and present their arguments and counter arguments over three rounds.

Simple Wiki Lastly, we use an encyclopedic text resource to also capture non-argumentative knowledge which can be relevant for explaining arguments. To save computational resources and increase the accuracy of the extraction process, we use the Wikipedia version in simple English.

Subtask	Measure	Al-Khatib	EREx
Relation Classification	macro F1	**0.79**	0.65
Relation Type Classification	macro F1	0.77	0.77
Identification of Concept 1	accuracy	0.69	**0.71**
Identification of Concept 2	accuracy	0.28	**0.35**

Table 4: Effect relation extraction evaluation.

Both argumentative text resources mainly contain defeasible arguments. Thus, the effect relations which we extract from them and, consequentially, the effect graph should not be treated as facts.

After extracting the effect relations from text, we remove duplicates. We only consider an effect relation to be a duplicate, if it was extracted from the same sentence in the same resources twice, which most often happens because of citations. We intentionally keep effect relations that are identical except for the sentence they were extracted from because this might indicate that the effect relation is especially relevant.

For building the effect graph, we connect the extracted effect relations as follows: The lemmas of the subjects S and the objects O become nodes. We add one edge between S and O for every respective effect relation we extracted. Since we do not collapse the edges to not lose any information, the resulting graph is expected to contain multi-edges.

4 Evaluation

We evaluate the effect graph as follows: In section 4.1, we evaluate the effect relation extraction process using the subtasks defined by Al-Khatib et al. (2020). Then, we evaluate the extracted graph itself. In section 4.2, we compare the graph statistics. Afterwards, we evaluate both precision (section 4.3) and recall (section 4.4). In this context, precision expresses the chance that a randomly selected edge of the graph is correct. We consider a statement to be correct if it is in accordance with the statement it was extracted from. Recall on the other hand is meant to measure the chance that a given effect relation is contained in the graph.

Baselines For the evaluation of the extraction subtasks defined by Al-Khatib et al. (2020), we use their models as a baseline, denoted by **Al-Khatib**. For evaluating the effect graph as a whole, we build the effect graph as described in section 3.2, but using different extraction methods. We use the **OpenIE** implementation which is part of Stanford CoreNLP (Manning et al., 2014; Angeli et al., 2015) to extract subject-verb-object triples, applying a confidence threshold of 0.9. We accept such triples as effect relations where the verb is an effect word and the subject and object link to Wikipedia pages. Further, we use a version of EREx where we do not require the subject and object to link to Wikipedia, denoted by **EREx***. We expect this version to have a higher recall, but also more noise.

4.1 Extraction Subtasks

Al-Khatib et al. (2020) propose several subtasks for effect relation extraction. These subtasks include:

- **Relation Classification:** Classify whether a statement does contain an effect relation;
- **Relation Type Classification:** Predict the effect relation's polarity;
- **Identification of Concept 1:** Identify the effect relation's subject;
- **Identification of Concept 2:** Identify the effect relation's object.

For the first two subtasks, Al-Khatib et al. (2020) propose a supervised model, while for the last two they rely on the OpenIE approach of Stanovsky et al. (2018). To make the comparison fair, we slightly adopt EREx such that it predicts a relation type and identifies concepts even if it does not detect an effect relation. For the evaluation, we use the dataset published by Al-Khatib et al. (2020), which contains crowd annotations for the different subtasks, and compare our results to the results reported in their paper.[2] The results are presented in table 4.

Concerning Relation Classification, EREx misses effect relations considerably more often than it wrongly predicts one (1582 vs 174 instances), which fits our focus on precision rather than recall. When counting only such instances

[2]As the train-test-split used by Al-Khatib et al. (2020) is unknown to us, we use the full dataset for the evaluation. Thus, unfortunately, the results are not directly comparable.

Dataset	Number of effect relations		
	EREx	EREx*	OpenIE
Debatepedia	1.6k	8.8k	9.9k
Debate.org	150.3k	669.9k	1173.8k
Simple Wiki	43.6k	193.9k	290.3k

Table 5: Effect relation extraction statistics

	EREx	EREx*	OpenIE
# Nodes	53k	734k	129k
# Edges	195k	872k	1474k
# Positive edges	157k	729k	1250k
# Negative edges	38k	142k	223k
# Connected node pairs	126k	733k	603k

Table 6: Effect graph statistics.

where EREx extracts a relation, it correctly detects its polarity in 85%, the subject in 80% and the object in 41% of the instances. While both models' scores of identifying the object are low, this can be explained at least partly by the measure: The object is considered to be wrong if it is off by one word, even if it is an article. In the dataset, it is inconsistent whether articles are part of the object or not.

4.2 Graph Statistics

Table 5 shows the number of edges, i.e., extracted effect relations, per dataset. Table 6 contains some basic statistics of the effect graph. The number of connected node pairs is included because of the high ratio of multi-edges. We consider (A,B) and (B,A) as the same node pair. Table 7 shows the number of overlapping nodes between the different effect graph versions.

Overall, using OpenIE results in the largest graph and using EREx in the smallest. That OpenIE extracts fewer nodes than EREx* is likely due to the required linking to Wikipedia. For all three methods, there are considerably more positive than negative effect relations.

4.3 Precision

As the effect graph is generated by extraction from large text resources, we do not have a ground truth of whether or not a statement was extracted correctly. Thus, we evaluate precision a posteriori. For this purpose, we randomly select 250 edges per graph. For each, we annotate whether it was extracted correctly, given the original statement (*yes, rather yes, unsure, rather no, no*). We both do an expert annotation by one of the authors and crowd annotations via mturk.

	EREx	EREx*	OpenIE
EREx	–	52,821	43,527
EREx*	52,821	–	63,827
OpenIE	43,527	63,827	–

Table 7: Effect graph: Node overlap.

Instructions

We require the crowd workers to successfully pass an instruction before working on the task. The instruction consists of a short description of the task, two examples with comments, three instances which had to be annotated correctly, and an optional field where the workers could write comments. The description, examples and the first instance are provided in appendix A.

Overall, the task should be as intuitive as possible. For this purpose, we did not show the concrete verb of the effect relation, but just the effect's polarity. Instead of explaining that we are not interested in modality, we framed the polarity as "(may) negatively affect". We addressed the risk of confusion with sentiment by addressing it in the instructions: Though most would likely agree that *ending war* is desirable, we highlight that the effect which is expressed on *war* is a negative one. The workers then have to correctly identify two further such effects as negative (*coal power* reducing *CO2-emissions*) respectively positive (*current EU policy* leading to a *financial crisis*). Similarly, we exemplify and control that the subject and object have to be identified correctly.

Annotation Process

We only accept workers who live in the US and have a HIT approval rate greater than 98% and more than 10,000 approved HITs in total. Additionally, they have to have passed the instructions with three correct answers out of three. As the cases in the instructions were not ambiguous, we count *rather yes* and *rather no* as wrong answers, as well as *unsure*. Overall, only 9 out of 50 workers passed the instructions.

We have a total of 750 instances to be annotated. Each instance is annotated by three crowd workers and one expert. Overall, seven of the nine qualified workers did actually address the task. Of these seven workers, three did annotate the vast majority of the instances (747, 739 and 650 respectively).

categorial label	value
yes	2
rather yes	1
unsure	0
rather no	−1
no	−2

Table 8: Mapping categorial answers to values.

		crowd	expert
polarities	Fleiss	0.15	0.26
	Randolph	0.47	0.44
scalar	Krippendorff	0.20	0.34
	Pearson		0.57
	Spearman		0.56

Table 9: Agreement scores for effect relation evaluation.

Agreement

We treat the five labels either as *polarities*, mapping *rather yes* to *yes* and *rather no* to *no*. Or we treat them as *scalars* as indicated in table 8. The mapping allows us to intuitively combine multiple labels by computing their mean. This is relevant later for generating the label to ultimately measuring the precision. But it also enables us to measure the agreement between the combined label and the expert annotator (*expert*). Additionally, we compute the agreement among the crowd workers (*crowd*). For mapping back from numbers to labels, we always round up positive values and round down negative values. This way, the labels *yes* and *no* are only provided if there are no opposing polarities and the label *unsure* is given as rarely as possible.

We use the following agreement scores: **Fleiss Kappa** for categorial agreement respecting the label distribution; **Randolph Kappa** (Randolph, 2005) for categorial agreement without respecting the label distribution; **Krippendorff Alpha** (Krippendorff, 2011) for scalar agreement, especially in the *crowd* setup as it allows for multiple annotators; **Pearson Correlation** for scalar agreement in the *expert* setup, using the mean as is; **Spearman Correlation** for rank agreement in the *expert* setup, mapping the mean to labels.

The scores are presented in table 9. Overall, the agreement is rather weak. Concerning *polarities*, we note two things: First, there is a big difference between *Fleiss* and *Randolph* which can be explained by the fact that the crowd workers tended to annotate *yes* or *rather yes* way more often than *no* or *rather no* . Second, for Fleiss, the involvement of the expert leads to higher scores, while for Randolph it is vice versa. This tendency might be explained by the fact that the expert annotated *yes* or *rather yes* even less often than *no* or *rather no*. So the expert reduces the imbalance between these two labels which in turn causes Fleiss and Randolph to approach each other.

For the scalar agreement, the scores are a bit better which makes sense as only in this scenario the labels' ranks are considered properly. However, we still conclude that the agreement is weak which we have to consider when interpreting the results.

Results

The precision scores are calculated by dividing the number of correctly extracted effect relations by the sum of the numbers of correctly and incorrectly extracted ones. As for what we consider a correctly extracted effect relation, we again consider different settings to provide a full picture. For one, we use either the expert label or the aggregated crowd label. Further, we either consider only the labels we are confident about, namely *yes* and *no* (denoted by *exclusive*), or we again aggregate *yes* and *rather yes* as well as *no* and *rather no* (denoted by *inclusive*). We never consider the relatively few cases where the (aggregated) label is *unsure*. The results are shown in table 10.

The expert's tendency to annotate *yes* considerably less often than the crowd workers is reflected by the overall lower precision scores. Despite this large difference of the scores, the tendency among the datasets is consistent for the crowd workers' and the expert's annotations: EREx and EREx* clearly outperform OpenIE, while EREx seams to be at least slightly better than EREx*. This was to be expected as EREx is more restrictive in selecting subjects and objects than EREx*.

We conclude that EREx and EREx* are most likely more precise than the OpenIE baseline, but whether or not they are precise enough for our envisioned use case is yet to be shown.

4.4 Recall

For evaluating recall, we check whether the graph does contain such effect relations which we would expect it to contain. In order to do so, we build an evaluation dataset. We choose one random argumentative claim per topic from the Debatepedia dataset of arguments related to consequences

	Crowd Annotations				Expert Annotations			
	exclusive		inclusive		exclusive		inclusive	
	total	precision	total	precision	total	precision	total	precision
OpenIE	115	0.83	237	0.70	186	0.38	241	0.34
EREx	132	**0.98**	246	**0.80**	174	**0.54**	243	**0.54**
EREx*	130	0.95	242	0.79	175	0.48	248	0.46

Table 10: Effect graph precision.

(Kobbe et al., 2020). This results in 180 claims. From each claim, we manually extract all effect relations which we consider reasonable. This results in 308 effect relations. If there is more than one possible effect relation for a claim, we annotate whether they are either equivalent to ($\equiv$), disjoint to ($\not\equiv$), or part of ($\supset$) the other ones. Table 11 shows some examples which we will briefly discuss.

In example 1, there exist three reasonable effect relations which differ only in the concreteness of the object, *a* being the most concrete and *c* the least. Note that the effect verb *eliminate* is only correct when mentioning the *ability* of restaurants. Still, the statement indirectly also expresses that *calorie counts* negatively effect restaurants, which is why in effect relation *c*, there is no effect verb annotated. Example 2 briefly shows a case where there exist two effect relations which are roughly equivalent in terms of the information they contain. In contrast, in example 3 exist two completely distinct effect relations, though the second one is rather implicit. Example 4 is a bit more complex: *a* is as concrete as possible, but it can be split in *b* and *c* which together are equivalent to *a*.

For calculating recall, we use two straightforward formulas: We either divide the number of the ground truth effect relations which are contained in the effect graph by the total number of ground truth effect relations (*total*), or we divide the number of claims for which at least one ground truth effect relation is contained in the effect graph by the number of claims in the dataset (*per statement*). Further, we optionally exclude the effect relations which were extracted from Debatepedia from the effect graph (*w/o DP*). Though it is unclear what results one can expect this way, we consider it to be a purer way of calculating recall.

The results (see table 12) show a clear trend: EREx has lower recall than OpenIE, while EREx* has a significantly higher recall than OpenIE only when Debatepedia is included in the graph. Importantly, we note that EREx* is only better than EREx in the *full graph* setting. This fits our observation that the effect relations extracted by EREx* tend to be overly specific oftentimes, which is one reason why we proposed the linking to Wikipedia as an additional requirement.

As the recall is particularly low for the settings without Debatepedia, we take a brief look at the few successes in table 13: It is noticeable though unsurprising that the graphs generated with EREx and EREx* contain the exact same test instances. Further, two of them (7,8) are not identified by OpenIE which in turn contains seven instances which EREx and EREx* do not (9-15). One of the latter instances cannot be included in EREx or EREx* because it contains a non-nounphrase as subject (14) – but considering the unspecificity of instance 14, this restriction seems to be justifiable.

5 Explanation Generation

For generating explanations, we use the effect graph generated by EREx. As outlined in the introductory section, we envision two different types of explanations which we will describe separately in the sections 5.1 and 5.2. Afterwards, we introduce a measure to rank the potential explanations (5.3).

5.1 Effect-Effect-Explanation

For an Effect-Effect-Explanation to be meaningful, the polarities have to fit the relation we aim to explain. Concretely, we explain a positive relation either by two positive or two negative relations, and a negative relation by combining a positive and a negative one. To generate explanation candidates , we use the effect graph in a straight forward way by querying for paths of length two between the instances of interest with appropriate edge polarities. As a result, we get a list of explanation candidates.

For explaining how abortions protect women, this list includes 370 explanation candidates, though many of them are similar to each other because of our loose definition of duplicates. In-

Ex. 1	**Calorie counts eliminate ability of restaurants to be spontaneous.**
a	(Calorie counts) [-eliminate] (ability of restaurants to be spontaneous)
b	$\supset$ (Calorie counts) [-eliminate] (ability of restaurants)
c	$\supset$ (Calorie counts) [-] (restaurants)
Ex. 2	**Circumcision creates risk of infections in infants**
a	(Circumcision) [+creates] (risk of infections)
b	$\equiv$ (Circumcision) [+creates] (infections)
Ex. 3	**Assassinations protect publics from terrorism; even while it's hard to measure**
a	(Assassinations) [+protect] (publics)
b	$\not\equiv$ (Assassinations) [-protect from] (terrorism)
Ex. 4	**Network neutrality damages competition and niche suppliers**
a	(Network neutrality) [-damages] (competition and niche suppliers)
b	$\equiv$[(Network neutrality) [-damages] (competition)
c	$\not\equiv$ (Network neutrality) [-damages] (niche suppliers)]

Table 11: Examples: Effect relation annotation for recall evaluation.

	total		per statement	
	full	w/o DP	full	w/o DP
OpenIE	0.07	**0.04**	0.14	**0.09**
EREx	0.05	0.03	0.09	0.06
EREx*	**0.14**	0.03	**0.28**	0.06

Table 12: Effect graph recall.

	EREx + EREx* + OpenIE
1	icc $\xrightarrow{-}$ crimes
2	abortion $\xrightarrow{+}$ women
3	eating meat $\xrightarrow{-}$ animals
4	marijuana $\xrightarrow{-}$ productivity
5	war $\xrightarrow{-}$ civilians
6	affirmative action $\xrightarrow{-}$ meritocracy
	EREx + EREx*
7	two-state solution $\xrightarrow{+}$ stability
8	gay marriage $\xrightarrow{-}$ procreation
	OpenIE
9	elections $\xrightarrow{+}$ judges
10	government $\xrightarrow{+}$ public transport
11	stimulus $\xrightarrow{+}$ debt
12	circumcision $\xrightarrow{+}$ infections
13	primaries $\xrightarrow{+}$ candidates
14	they $\xrightarrow{+}$ headaches
15	rights $\xrightarrow{+}$ contracts

Table 13: Effect graph recall (*w/o DP*): Successes.

stead of listing all candidates, we list all the interim nodes C used within the explanation candidates: *, choice, country, fetus, god, man, nothing, order, people, person, pregnancy, right, sex, society, t, unwanted pregnancy, woman 's rights. One can easily imagine that some of the concepts mentioned are useful for explaining why abortions protect women, while others are non-sense.

5.2 Effect-Lexical-Explanation

Sometimes, we need additional lexical knowledge for explaining an effect relation. As mentioned previously, we use WordNet to incorporate some of the potentially relevant lexical knowledge. Concretely, this includes hyperonymy, meronymy and synonymity.

To extract explanation candidates for $A \xrightarrow{\pm} B$, we again look for instances C, considering the following cases: $A \xrightarrow{\pm} C \xrightarrow{\text{WN}} B$ and $A \xrightarrow{\text{WN}} C \xrightarrow{\pm} B$. The polarities have to be identical and $\xrightarrow{\text{WN}}$ indicates one of the lexical relations mentioned above.

For the example, we find 10 different explanation candidates. Half of them argue that abortions are good for mothers in some way, and mother is a hyponym for woman. While being trivial, we still think that there is a benefit in this explanation. It states correctly that the positive effect of the abortion is on the mother (and not on the fetus, for instance) and finds the relation between *mother* and *woman*. The other five explanation candidates use the interim nodes *people, action, failure, man* and none of these explanations seems useful to us.

5.3 Explanation Candidate Ranking

Since the proposed methods to generate explanations often result in a list of explanation candidates of varying quality, we further propose a simple means of ranking them which is inspired by tf-idf. The idea is to measure the importance of the interim node C based on its degree in the effect graph (denoted by deg_e), where we assume a lower degree to be better as it indicates specificity, and its degree in the subgraph connecting A and B (denoted by deg_s), where we consider a higher degree to indicate relevance. The core idea for measuring importance is the quotient of these two quantities. This quotient, however, does not respect the absolute quantities and will thus lead to the same score for C having degree 1 in both graphs and having degree 5 in both graphs, though we consider the latter to be considerably better. In order to account for that, we apply the idea of additive smoothing and increment the denominator by 1. Further considering that we rather prefer medium in- and out-degree rather than a high (low) in- and low (high) out-degree, we calculate C's importance for Effect-Effect-Explanations as follows:

$$\frac{indeg_s(C)}{indeg_e(C)+1} \cdot \frac{outdeg_s(C)}{outdeg_e(C)+1}$$

Considering Effect-Lexical-Explanations, we are only interested in either C's out- or in-degree. For better comparability, we use the square of the relevant quotient to measure the importance.

When applying the importance measure on the example, the five most important nodes are in descending order: unwanted pregnancy, woman 's rights, mother, fetus, pregnancy. The corresponding explanation via *unwanted pregnancy* unfortunately does not make sense due to an extraction mistake, although the concept seems to be ranked that high for good reason. We already discussed the one via *mother* in section 5.2. The others suggest that abortions kill fetuses which in turn harm, damage or endanger the woman; that abortions end pregnancies which also harms the woman; and that abortions support women's rights which in turn are good for women.

6 Conclusion

We propose a method to extract effect relations from text and use it to build an effect graph. We further propose a method to use the effect graph as background knowledge for automatically generating structured explanations, for example for arguments from consequences. However, the effect graph's precision remains unclear while its recall is low. The latter issue might be addressed by either improving the extraction method or, to a certain degree, by running the method on larger text resources. The effect graph can be seen as a valuable resource on its own, as it can potentially be used to also address other tasks than explanation generation, like identifying (counter-) arguments for a specific topic or extending common sense knowledge graphs such as ConceptNet (Speer et al., 2017).

Limitations

While the proposed methods are attractive due to their efficiency, explainability and not needing training data, the limitations are also manifold: The pipeline nature propagates all errors that occur. For instance, the dependency parser in use performs rather poorly on informal texts such as tweets. Further, our definition of positive and negative effect relations is quite shallow and does not always live up to the real world's complexity. We only capture effect relations that are formulated explicitly within one sentence, and only one effect relation per sentence. Requiring the nodes to link to Wikipedia might be too restrictive while not even truly solving the problem of filtering non-sense nodes. Both the low inter-annotator-agreement in our effect graph evaluation as well as the discrepancy of the crowds' and the expert's annotations make it hard to assess the correctness of the extracted effect relations. And lastly, while we showcase some generated explanations, we did not properly evaluate how reliable the approach is in finding reasonable explanations. Indeed, first results suggest that this approach of generating explanations works rather inconsistently, though the ranking helps to a certain degree.

What one might consider another limitation is that we do not check the effect relations for factual correctness, which ultimately leads to contradictions and inconsistencies in the effect graph. While fact checking is a difficult and controversial task, we also purposefully decided against any form of fact or consistency checking. Each edge in the effect graph is meant to represent one effect relation exactly as it was expressed. Including critical effect relations in the graph allows for identifying, analyzing, and potentially disproving them.

Acknowledgments

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project ExpLAIN, Grant Number STU 266/14-1, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

References

Khalid Al-Khatib, Yufang Hou, Henning Wachsmuth, Charles Jochim, Francesca Bonin, and Benno Stein. 2020. End-to-end argumentation knowledge graph construction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7367–7374.

Khalid Al Khatib, Lukas Trautner, Henning Wachsmuth, Yufang Hou, and Benno Stein. 2021. Employing argumentation knowledge graphs for neural argument generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4744–4754, Online. Association for Computational Linguistics.

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: Long Papers)*. Association for Computational Linguistics.

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.

Donald Davidson. 1967. Causal relations. *The Journal of Philosophy*, 64(21):691.

Esin Durmus and Claire Cardie. 2018. Exploring the role of prior beliefs for argument persuasion. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1035–1045. Association for Computational Linguistics.

Esin Durmus and Claire Cardie. 2019. A corpus for modeling user and language effects in argumentation on online debating. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Christiane Fellbaum. 2010. Princeton university: About wordnet.

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. Association for Computational Linguistics. Event-place: Copenhagen, Denmark.

Roxana Girju. 2003. Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, page 76–83, USA. Association for Computational Linguistics.

Roxana Girju and Dan Moldovan. 2002. Text mining for causal relations. In *FLAIRS conference*, pages 360–364.

Stefan Heindorf, Yan Scholten, Henning Wachsmuth, Axel-Cyrille Ngonga Ngomo, and Martin Potthast. 2020. Causenet: Towards a causality graph extracted from the web. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3023–3030, New York, NY, USA. Association for Computing Machinery.

Jonathan Kobbe, Ioana Hulpuș, and Heiner Stuckenschmidt. 2020. Unsupervised stance detection for arguments from consequences. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 50–60, Online. Association for Computational Linguistics.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Jose L. Martinez-Rodriguez, Ivan Lopez-Arevalo, and Ana B. Rios-Alvarado. 2018. OpenIE-based approach for knowledge graph construction from text. *Expert Systems with Applications*, 113:339–355.

Justus J Randolph. 2005. Free-marginal multirater kappa (multirater k [free]): An alternative to fleiss' fixed-marginal multirater kappa. *Online submission*.

Hannah Rashkin, Sameer Singh, and Yejin Choi. 2016. Connotation Frames: A Data-Driven Investigation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 311–321. Association for Computational Linguistics. Event-place: Berlin, Germany.

Paul Reisert, Naoya Inoue, Tatsuki Kuribayashi, and Kentaro Inui. 2018. Feasible Annotation Scheme for Capturing Policy Argument Reasoning using Argument Templates. In *Proceedings of the 5th Workshop on Argument Mining*, pages 79–89, Brussels, Belgium. Association for Computational Linguistics.

Stephen Soderland, Brendan Roof, Bo Qin, Shi Xu, Mausam, and Oren Etzioni. 2010. Adapting open information extraction to domain-specific relations. *AI Magazine*, 31(3):93–102.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1):4444–4451.

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 (Long Papers)*, pages 885–895, New Orleans, Louisiana. Association for Computational Linguistics.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.

Jie Yang, Soyeon Caren Han, and Josiah Poon. 2022. A survey on extraction of causal relations from natural language text. *Knowledge and Information Systems*, 64(5):1161–1186.

Jian Yuan, Zhongyu Wei, Donghua Zhao, Qi Zhang, and Changjian Jiang. 2021. Leveraging argumentation knowledge graph for interactive argument pair identification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2310–2319. Association for Computational Linguistics.

A Part of the Crowd Workers' Instructions

Each HIT, you will be presented a **Statement** from which a **Relation** was extracted automatically. The Relation is expected to capture some sort of positive or negative effect between two of the statement's instances.
Your task is to judge whether the extraction was successful. Successful means that the Relation can be considered to be correct when assuming that the Statement itself is correct.

Example 1

Statement: Scientists found out that unicorns can end any war.

Relation: unicorns (may) **negatively** affect war

Obviously, the statement is made up. But for this task, we assume it to be true. Consequently, the Relation is **correct**: The effect which is expressed on *war* is a negative one (it may be *ended* by unicorns).

Other words that trigger negative effects are for instance *decrease, damage, forbid, ban, reduce, ...*
Positive effects are triggered by words such as *increase, help, permit, cause, create, ...*

Example 2

Statement: Scientists found out that unicorns can end any war.

Relation: scientists (may) **negatively** affect war

This time, the Relation is **not correct**: It is not the *scientists* who have a negative effect on *war*, but the *unicorns*.

Your turn!

Statement: Throughout history, nuclear weapons have killed many innocents.

Relation: history (may) **negatively** affect innocents

Assuming the Statement is correct, is the Relation also correct?

- ○ No
- ○ Rather no
- ○ I am unsure
- ○ Rather yes
- ○ Yes