Temporality in General-Domain Entailment Graph Induction

Anonymous ACL submission

Abstract

Entailment Graphs based on open relation extraction run the risk of learning spurious entailments (e.g. *win against* ⊨ *lose to*) from antonymous predications that are observed with the same entities referring to different times. Previous research has demonstrated the potential of using temporality as a signal to avoid learning these entailments in the sports domain. We investigate whether this extends to the general news domain. Our method introduces a temporal window that is set dynamically for each eventuality using a temporally-informed language model. We evaluate our models on a sports-specific dataset, and ANT – a novel general-domain dataset based on WordNet antonym pairs. We find that whilst it may be useful to reinterpret the Distributional Inclusion Hypothesis to include time for the sports news domain, this does not apply to the general news domain.

1 Introduction

The ability to recognise textual entailment and paraphrase is essential to many NLP applications, including open-domain question answering over unstructured data. This setting frequently poses the challenge that the answer to the question is not explicitly stated in the text, and can only be inferred using entailment rules and/or paraphrases. For example, the question might ask "Did Arsenal play Man United last night?" and the post-match report states "Arsenal beat Man United 1-0". A system that can recognise that *beat* ⊨ *play* will be able to provide the correct answer ("yes").

Entailment Graphs (Berant et al., 2011, 2015; Hosseini et al., 2018), learned using unsupervised methods applied over large text corpora, have been proposed as a means to support answering such questions. Entailment Graphs comprise nodes representing predicates, and edges representing the entailment relation between them. They can be learned using the Distributional Inclusion Hypothesis (DIH), which states that a predicate p entails a predicate q if the context set in which p can be used is *included in* the context set of q (Dagan et al., 1999; Geffet and Dagan, 2005). For predicates, the context set is usually interpreted as co-occurring argument pairs.

However, the argument pair-based, atemporal formulation of the DIH does not support the class of predicate pairs that are antonyms and occur frequently within a window of time with the same argument pairs, such as *winning* and *losing* (Guillou et al., 2020). For example, sports teams often play against each other multiple times in a season, likely with different outcomes, so that predicates such as *win against* and *lose to* are both likely to apply to the same sports team argument pairs (e.g. Arsenal, Man United). Consequently, current state-of-the-art methods for learning Entailment Graphs may commonly learn erroneous entailment relations between these pairs of highly correlated predicates (e.g. *to win* ⊨ *to lose*).

Guillou et al. (2020) propose an algorithm that circumvents this issue by considering argument pair occurrences only when the eventualities temporally overlap. This effectively reinterprets the DIH's context set as containing both argument pairs and time. They refine Entailment Graph induction for the sports news domain. We extend their work by applying the method to the general news domain, and propose setting different size temporal comparison windows for the different predicates contained in the general domain. We dynamically assign a different window size for each eventuality in the corpus using a temporally-aware language model (Zhou et al., 2020) that predicts the expected duration of the eventuality. We evaluate the Entailment Graphs on the Sports Entailment Dataset of Guillou et al. (2020), and ANT – a novel dataset derived from WordNet (Miller, 1995) antonyms.

We find that refining the DIH's context to include

time (in addition to argument pairs) is beneficial for the sports news domain, but that this does not extend to the general news domain. We do, however, identify predicates in legal news as another possible area in which temporal information may be useful for learning Entailment Graphs. Our contributions are: 1) the application of a temporally informed Entailment Graph learning method to the general news domain, and 2) ANT, a novel general-domain entailment dataset based on WordNet antonyms.

2 Background

2.1 Entailment Graphs

Entailment Graph Induction uses directional distributional similarity measures to determine whether an entailment relation holds between two predicates p and q. Successful measures include the purely directional Weed's precision score (Weeds and Weir, 2003), and the Balanced Inclusion score (BInc) (Szpektor and Dagan, 2008), which combines both symmetric and directional measures. We use these two scores as our baselines. Both scores are based on the Distributional Inclusion Hypothesis (DIH), which states that p entails q if the set of contexts in which p can be used is *included in* the context set of q (Dagan et al., 1999; Geffet and Dagan, 2005). When applying the hypothesis to predicates, the context set has mostly been taken to refer to argument pairs (e.g. by Berant et al. (2011) and Hosseini et al. (2018)).

Entailment Graphs have been built for a range of domains, including health (Levy et al., 2014), news (Hosseini et al., 2018), and commonsense (Yu et al., 2020). By focusing on the news domain we are able to leverage two sources of temporal information: the publication dates of the articles and the rich set of temporal expressions within them. Previous work has also considered a number of options for representing nodes in the graphs: typed binary predicates (Berant et al., 2011; Hosseini et al., 2018), Open-IE propositions (Levy et al., 2014), and eventualities (Yu et al., 2020). We use typed predicates, following Hosseini et al. (2018).

2.2 Temporality and Entailment Graphs

Guillou et al. (2020) incorporated temporal information into the graph learning framework of Hosseini et al. (2018), extending the local entailment score computation method to incorporate temporal filtering of eventualities. This reinterprets the DIH to include time in the context set of any given predicate. Unlike in Hosseini et al. (2018) where all eventualities of pairs of predicates that share the same arguments are considered for comparison, Guillou et al. (2020) aim to compare only those eventualities of predicates with shared arguments for which the underlying eventualities are temporally close to each other. The strength of this method is its ability to separate out instances of recurring eventualities, e.g. sports matches that occur between the same pair of teams. Following promising results for the sports domain, we extend the method to the general news domain.

2.3 Evaluating Entailment Graphs

Entailment Graphs are typically evaluated using datasets comprised of premise-hypothesis sentence pairs with labels denoting the entailment relation that holds between them. Dataset construction has been framed as a number of manual annotation tasks, e.g. image captioning (Bowman et al., 2015), question answering (Levy and Dagan, 2016), and fact verification (Schmitt and Schütze, 2019).

Evaluating entailments that involve temporality has received less attention. The *FraCas* test suite (Cooper et al., 1996) contains only a small number of temporal examples that based on entailments between predicates. *TEA* (Kober et al., 2019), which comprises sentence pairs in which temporally ordered predications have varying tense and aspect, does not include non-entailments that can be learned through the temporal separation of eventualities (e.g. outcome predicates *win* - *lose*). The *Sports Entailment Dataset* (Guillou et al., 2020) of entailment pairs between paraphrases of the predicates *play*, *win*, *lose*, and *tie*, was developed to address this gap. However, its narrow focus on sports makes it unsuitable for evaluating graphs for the general news domain. This motivates the construction of the general-domain ANT dataset.

2.4 Antonym Detection

Related to our work is the field of antonym detection, in which antonyms are distinguished from other semantic relations such as synonymy. We focus on the related but distinct task of Recognizing Textual Entailment (RTE) in the presence of antonymy, which can be seen as a more challenging version of the typical RTE setup. Antonymy detection is evaluated using various datasets, notably the relation classification-style EVALution dataset (Santus et al., 2015) and PPDB-based dataset of Rajana et al. (2017), and the multiple-choice GRE

question dataset (Mohammad et al., 2013). To compare our work to previous Entailment Graph models, we instead opt for the RTE paradigm, focusing on sentences containing binary predications. We note that the labels in ANT can easily be remapped for the evaluation of antonym detection systems.

3 Method

3.1 Relation Extraction

We start by extracting relation triples from a corpus of news articles. We use MONTEE (Bijl de Vroe et al., 2021), an open-domain system that uses the RotatingCCG parser (Stanojević and Steedman, 2019) and extracts relations consisting of predicates and their arguments by traversing the resulting CCG dependency graph. For each sentence we extract all potential *binary relations* of the form **arg1**-*predicate*-**arg2** (e.g. **Arsenal**-*beat*-**Man United**)[1]. Arguments, which may be either Named Entities or general entities (all other nouns and noun phrases), are mapped to their fine-grained FIGER types (Ling and Weld, 2012) (e.g. PERSON, DISEASE, etc.).

We extended MONTEE to add temporal intervals to binary relations where there is a path in the dependency graph between the predicate and a temporal expression in the text. The temporal intervals consist of the start and end date of the eventuality, and are derived using SUTime (Chang and Manning, 2012) – a tool for automatically identifying and resolving temporal expressions (such as "Monday 7th March 2022") found in the text, to a calendar date range. Expressions such as "yesterday" are resolved relative to the article's publication date.

3.2 Graph Learning with Temporal Filtering

To learn Entailment Graphs we use the temporal filtering method of Guillou et al. (2020) which extends the graph learning framework of Hosseini et al. (2018). The input is the set of typed binary relations paired with their time intervals. The output is a set of graphs, one for each pair of FIGER types found in the set of binary relations. We focus on locally learned entailments, leaving an investigation of the interaction between temporality and globalisation to future work.

In Hosseini et al. (2018) similarity scores between predicates are computed over feature vectors, with one feature vector per typed predicate. The feature in the vector is the argument pair from the binary relation (e.g. Arsenal, Man United) and the value is the pointwise mutual information (PMI) between the predicate and argument pair. Guillou et al. (2020) add a method to filter the counts of predicate p according to whether each eventuality's time interval overlaps with any of q's. That is, an eventuality in p is retained (and counted) if it is temporally close enough to any eventuality in q. The goal of this process is to separate out different instances of recurring eventualities involving the same argument pairs.

play	Arsenal	Man United	18/1/2021
beat	Arsenal	Man United	18/1/2021
play	Arsenal	Man United	12/2/2021
lose to	Arsenal	Man United	12/2/2021

Pair	play-beat	beat-play	play-lose	lose-play	beat-lose	lose - beat
Regular	2	1	2	1	1	1
Filtered	1	1	1	1	0	0

Figure 1: **Above:** Two sports matches between the same teams. **Below:** Regular and temporally filtered counts.

For example, suppose two football matches are held between Arsenal and Man United, one described as happening on 8th January 2021 where "Arsenal *played* and *beat* Man United.", and another on 12th February 2021 where "Arsenal *played* and *lost to* Man United" (see the upper section of Figure 1). The algorithm computes a *filtered count* for each argument pair for the pair *p-q*: the total number of eventualities of predicate p with a time interval that temporally overlaps with the time interval of any eventuality of predicate q, and vice versa. In this case the filtered count for *play-beat* = 1 and *play-lose to* = 1 as there is a temporal overlap for the *play* and *beat* events in the first match and the *play* and *lose to* events in the second. Crucially, *beat-lose to* = 0 as there is no temporal overlap between the *beat* and *lose to* events, which occurred on different days. See Figure 1 for an illustrated example and Guillou et al. (2020) for further details. We use the filtered counts to compute the temporal similarity measures described in Section 3.4. The regular, unfiltered, counts are used to compute their (standard) non-temporal counterparts.

Following completion of the temporal filtering process for all predicate pairs, we learn the following entailment relations: *beat* $\models$ *play*, *lose to* $\models$ *play*, and *lose to* $\not\models$ *beat* (and its reverse). Without

[1] As we are not concerned with the intersection of temporality and modality, we do not tag relations for modality.

temporal filtering a spurious entailment relation between *beat* and *lose to* (and vice versa), which occur within a similar context (i.e. they share the same argument pair), would be learned.

3.3 Dynamic Temporal Window

Although a uniform temporal window is suitable for sports matches, which are typically concluded within a single day, it may be less suitable for other eventualities. We follow the recommendation of Guillou et al. (2020) and apply a dynamic window on a per-predicate basis to reflect that different eventualities remain relevant for different lengths of time. For example, the window around information stating that a person *is president* should be larger than a report of a person *visiting a location*.

We incorporate a temporally-aware language model, *TacoLM* (Zhou et al., 2020), and use it as the basis for per-predicate dynamic windowing. TacoLM predicts the expected duration of an eventuality using the context provided by the sentence in which the eventuality mention occurs. For each eventuality in a sentence it assigns a duration label from the set $\{seconds, minutes, hours, days, weeks, months, years, decades, centuries\}$. In a small number of cases TacoLM is unable to make a prediction, indicated by the *no_prediction* label[2].

In the uniform window model, each eventuality e is assigned a temporal interval $e_t = [t_{start} - w, t_{end} + w]$, where t_{start} and t_{end} are predicted using SUTime(e), and w is the model's fixed window size. In the dynamic window model, we instead assign $e_t = [t_{start} - map(TLM(e)), t_{end} + map(TLM(e))]$. Here $map(TLM(e))$ is TacoLM's prediction mapped to a concrete duration value: $\{seconds, minutes, hours, days\} \mapsto 5$, $weeks \mapsto 15$, $months \mapsto 30$, $years \mapsto 365$, $decades \mapsto 3,650$, $centuries \mapsto 36,500$. That is, for shorter durations we maintain a uniform window of 5 days, extending it only for eventualities with longer durations.

3.4 Similarity Measures

We compute both a symmetric and a directional temporally-informed similarity measure to learn entailments, making use of the temporally filtered counts and PMI scores described in Section 3.2. We adapted BInc (Szpektor and Dagan, 2008) and Weeds' precision (Weeds and Weir, 2003).

Temporal Weed's precision: Weed's precision is computed using the temporally-filtered counts.

Temporal BInc-based measure: As a proxy to computing Conditional PMI between an argument pair, predicate p, and predicate q, which would be computationally expensive (if not infeasible) given the existing graph construction framework, we scale the original PMI scores. The temporally filtered $PMI_t = PMI \cdot (c_t/c)$, i.e. the original PMI multiplied by the ratio of filtered counts (c_t) to regular counts (c). We refer to this measure as T. Binc (Ratio PMI).

4 Evaluation

We evaluate the Entailment Graphs using two different entailment datasets. 1) the Sports Entailment Dataset (Guillou et al., 2020) which contains 1,312 entailment pairs, focusing on events that occur between two sports teams. 2) ANT, a novel dataset based on WordNet antonym pairs. ANT addresses the need for a *general-domain*, RTE-style dataset containing antonyms.

4.1 ANT Dataset Construction Overview

ANT[3] contains entailment pair examples of the form *premise, hypothesis, label*. The premise and hypothesis take the form of natural English sentences containing a subject, predicate, and object. The label denotes one of four types of entailment relation: 1) *Antonym*: non-entailments between antonymous predicates (e.g. *acquit - convict*), 2) *Directional Entailments* an antonymous predicate and a related third predicate (e.g. *acquit* $\models$ *indict*), 3) *Directional Non-Entailments*, the reverse of each Directional Entailment (e.g. *indict* $\not\models$ *acquit*), and 4) *Paraphrases* of each predicate in the antonym pair (e.g. *acquit - absolve*). For a standard entailment evaluation setup, we map: $(Antonyms, Dir.Non\text{-}Entailments) \mapsto 0$ and $(Paraphrases, Dir.Entailments) \mapsto 1$. Our released dataset contains the original four labels as these may be useful in future research.

Dataset construction was semi-automatic. The manual steps were carried out by two expert annotators: one native, and one fluent English speaker[4]. Our dataset generation method uses the entailment relations between manually annotated predicate *clusters* to generate entailment pairs. By ensuring that most of the annotation occurs at the *predicate*

[2] 249,262 [0.61%] eventuality mentions in the NewsSpike corpus

[3] https://anonymous-link.com

[4] Both annotators were authors of this paper

level, rather than the *predicate-pair* or *sentence-pair* level, we are able to generate thousands of high quality entailment pairs from hundreds of annotated predicates. This is in contrast with the construction processes of the Levy (Levy and Dagan, 2016) and SherLIiC (Schmitt and Schütze, 2019) datasets, which involved generating large numbers of candidate entailment pairs of varying quality, prior to manual annotation by crowd-source workers. Our method also avoids the issue of selection bias present in Zeichner et al. (2012) and SherLIiC, that arises from using a similarity measure to automatically pre-select candidate entailments.

4.2 Antonym Pair Selection

We started by automatically collecting a list of 477 lemmatised verb antonym pairs from WordNet (Miller, 1995) and propose these as possible conflicting predicate pairs. Although WordNet's antonym set is not large, the high quality of its annotations makes WordNet a reliable starting point.

We excluded antonym pairs that express a type of temporal entailment (e.g. *fall asleep* and *wake up*), as these appear to express a more complicated relationship than simple antonymy. While these predicate pairs are antonymous when interpreted as simultaneous eventualities, they also entail each other at some temporal distance (e.g. you cannot *fall asleep* and *wake up* at the same time, but you need to *fall asleep* before you can *wake up*). If one of the two human annotators marked the antonym pair as having a possible temporal entailment between the predicates, we removed it from the set. This step resulted in 283 remaining antonym pairs.

We also removed pairs that were highly specific (e.g. *dehydrogenate-hydrogenate*) as these are likely to be infrequent in the general domain, pairs resulting from simple alternation of prepositions or morphemes (*scale up-scale down*; *deceive-undeceive*), and duplicate pairs in the British spelling.[5] We were left with 114 antonym pairs.

4.3 Entailment Cluster Construction

For each antonym pair, we identified possible paraphrases and third predicates that are entailed by both. We used the online Merriam-Webster Thesaurus (Merriam-Webster, 2021), which includes both (near) synonyms and antonyms, and the Relatedwords website (RelatedWords, 2021) – an online tool for finding related words beyond synonyms, which combines a number of NLP resources including word embedding spaces, ConceptNet and WordNet. This helped us find less typical paraphrases and often suggested entailed predicates.

For each antonym pair we created an *entailment cluster* $\mathcal{C} = (A_1, A_2, E)$, where A_1 and A_2 are the sets of predicates containing the *first* and *second* predicate in the seed antonym pair respectively, plus their paraphrases, and E is a set of predicates entailed by all the elements in $\cup(A_1, A_2)$.

Each cluster was then manually annotated with a set of argument type pairs (distinct from the FIGER types for Named Entities), which were later used for instantiating simple sentences. For example, the cluster for the antonym seed pair *refresh-tire* receives a set containing a single argument type pair, *activity#generic_person*. We allowed predicates with a specific word sense to be assigned a specific set of types. For example, for the *enjoy-suffer through* pair, the entailed predicate *see* is assigned the set containing just the type *generic_person#entertainment_watch*, to avoid it being paired with arguments from the *entertainment_read* type. This also enabled us to specify argument order, allowing a predicate pair like *refresh(activity#generic_person) - do(generic_person#activity)*.

4.4 Entailment Pair Generation

The aim of the generation step is to automatically convert the entailment clusters into the dataset format required for evaluation: premise, hypothesis, and a label denoting the type of entailment relation that holds between them.

To generate entailment pairs we take the cross product of different sets in the cluster. *Directional Entailments* are generated by $\cup(A_1 \times E, A_2 \times E)$, *Antonyms* by $\cup(A_1 \times A_2, A_2 \times A_1)$, *Directional Non-Entailments* by $\cup(E \times A_1, E \times A_2)$ and *Paraphrases* by $\cup(A_1 \times A_1, A_2 \times A_2)$, excluding duplicate predicates. We exclude an entailment pair if no intersection is found in the sets of its argument types, or if it already occurs as part of another antonym pair's cluster.

To generate a sentence for a predicate we need to populate its subject and object arguments. We therefore manually created argument strings for each argument type, ensuring they combine effectively with all predicates in the cluster. For example, the argument type *politician* maps to ar-

[5] We prefer American English spellings (e.g. *colonize*) over British English spellings (*colonise*) as the training corpus contains mostly American English news articles.

guments like *Hillary Clinton*, used to instantiate sentences for predicates like *govern*. We used the Relatedwords website (RelatedWords, 2021) for inspiration. We then sampled an argument type pair from the intersection of those that apply for both predicates in the entailment pair. For each argument type we sampled non-identical argument strings. This produces an entailment example of the form (*arg1, predicate1, arg2. arg1, predicate2, arg2. label*). For example, (*The school, admitted, Jean. The school, evaluated, Jean. 1*) represents the directional entailment *admit* ⊨ *evaluate*. Finally, both annotators made a single pass over the dataset to identify errors, and corrected the clusters accordingly. For example, they encountered unforeseen predicate-argument mismatches stemming from word sense ambiguity. Whilst this refinement method may be repeated indefinitely, we found that after a single manual pass the quality of the generated sentence pairs was very high.

The test portion[6] of ANT (based on 100 WordNet antonym pairs) contains 6,300 entailment pairs: 1,800 Antonyms, 1,465 Directional Entailments, 1,465 Directional Non-Entailments, and 1,570 Paraphrases. For the purpose of evaluation we used the following data subsets: 1) **Base**: *Antonyms* and *Directional Entailments*, and 2) **Directional**: *Antonyms* and *Directional Non-Entailments*.

4.5 Error Analysis

To verify the dataset's quality we conducted an error analysis on 200 examples, with 50 examples per label sampled randomly from the test set. We found 82.5% (165 /200) examples to be *correct*, confirming that the dataset is of high quality. Of the 35 *incorrect* examples we labelled five as a *syntactic error*, 18 as a *semantic error*, and 12 as *unnatural/disfluent*. The *syntactic errors* were attributed to wrong verb tense or a missing auxiliary verb in the predication. Sometimes *semantic errors* resulted from the introduction of subtle meaning change, such as for the directional non-entailment "Morgan **changed** the server" - "Morgan upgraded the server" (here **changed** might be interpreted as **replaced**). They also arose due to predicate pairs that were overlooked in cluster construction, e.g. **look down on** is an antonym of **like** but not necessarily a paraphrase of **dislike** - you can **dislike** (a person) without **looking down on** (them). *Unnatural* sentences were often the result of odd argument-predicate combinations, e.g. "Gale **expended** gas".

[6] ANT also contains a small development set (based on 14 antonym pairs) for use with supervised learning techniques

5 Experimental Setup

We used the NewsSpike corpus of multi-source news text (Zhang and Weld, 2013) for all of our experiments. NewsSpike comprises approx. 0.5M articles, collected over a period of 6 weeks.

Using MONTEE, we extracted 40,669,470 binary relation triples from NewsSpike. Of these 8,107,944 (19.94%) binary relations are extracted with a temporal interval resolved by SUTime (Chang and Manning, 2012) from a temporal expression in the text. As the temporal filtering method relies on the information contained in the time intervals to compute the temporal overlap of two eventualities, the sparseness of temporal expressions in the text raises a problem. To address this we employ the strategy described in Guillou et al. (2020), using the SUTime temporal interval if it is available and backing off to the document publication date if not.

We used the *entGraph*[7] framework with the extension of temporal filtering by Guillou et al. (2020) to train the Entailment Graphs. See Appendix A for hardware requirements and parameter settings.

We conducted experiments using two main settings. For the sports domain we apply a uniform window of 5 days on either side of the temporal intervals. We chose this setting because the evaluation predicates all refer to sports matches. Since these have a short duration and occur frequently between different pairs of teams, the window for which a match stays relevant to the readers, and for which the preconditions and consequences of the eventuality hold, is typically short.

For the general domain the duration of eventualities is highly variable, ranging from minutes or hours, to years, decades, or even centuries. These eventualities may also remain relevant for much longer than the sports matches. We therefore apply a dynamic window around each time interval[8] (see Section 3.3 for details).

6 Results

Table 1 contains Area Under the precision-recall Curve (AUC) scores for the Base and Directional subsets of the Sports and ANT datasets. For the

[7] https://github.com/mjhosseini/entGraph

[8] We also investigated using a uniform window which led to slightly worse results

	Sports		ANT	
Data subset	Base	Dir.	Base	Dir.
Recall < *threshold*	0.75	0.75	0.3	0.3
Similarity measure:				
Weed's Pr (Count)	0.440	0.460	**0.181**	**0.199**
T. Weed's Pr (Count)	0.455	**0.472**	0.180	0.198
BInc (PMI)	0.471	0.432	0.161	0.178
T. BInc (Ratio PMI)	**0.495**	0.437	0.161	0.178
BInc (Count)	0.462	0.419	0.159	0.167
T. BInc (Count)	0.481	0.430	0.160	0.167

Table 1: AUC scores for the **Base** and **Dir**ectional subsets of the Sports Entailment and ANT datasets.

Sports subsets the temporal measures consistently outperform their non-temporal counterparts. In spite of the consistency across similarity scores, the difference is not statistically significant. For the Base and Directional subsets of ANT, performance of the temporal measures and their non-temporal counterparts is not significantly different. This suggests that the atemporal formulation of the DIH by Dagan et al. (1999) and Geffet and Dagan (2005) is appropriate for the general domain.

Figure 2 contains the precision-recall curves for the Sports Entailment and ANT datasets[9]. Each point on the curve represents a different entailment score threshold, with higher thresholds corresponding to lower recall, and vice versa. To provide a fair comparison between similarity scores that have different recall ranges, we compute AUC under a recall threshold, chosen separately for each dataset (See threshold values in Table 1). For the Sports Entailment Dataset we observe higher precision for the temporal measures compared with their non-temporal counterparts at the lower recall ranges.

For the ANT dataset we make two observations. Firstly, recall is very low. This is due to the absence of many of the entailment pairs in the Entailment Graphs. Secondly, in contrast to the Sports Entailment Dataset, the curves for the temporal and non-temporal measures are very similar, confirming that the temporal distributions for this domain's eventualities are such that temporal filtering has a negligible effect. This is further confirmed by the analysis presented in Table 2.

		True	False	$\delta(T-F)$
Sports	% Scaled	31.5	35.8	-4.2
	% Overlap	72.8	65.8	7.1
ANT	% Scaled	53.0	51.8	1.2
	% Overlap	50.4	50.4	0.0

Table 2: Analysing the difference in effect of temporal filtering between the Sports and ANT base datasets.

7 Analysis and Discussion

Table 2 contains statistics of temporal separation for the Base subset of the Sports Entailment and ANT datasets. *% Scaled* is the percentage of PMI scores (for each co-occurrence P, Q, AP of relations (P, AP) where P and Q are predicates, and AP is an argument pair) that are scaled down by the temporal filtering method. *% Overlap* is the percentage of eventuality comparisons (e_p, e_q) that result in a temporal overlap. When the method is effective, we expect *% Scaled* to be higher for false predicate pairs than true predicate pairs (as scores of antonymous predicate pairs should be scaled down). Scaling should be inversely related to the average *Overlap*, which we expect to be higher for true predicate pairs than false predicate pairs.

We indeed find that *% Scaled* is higher for false predicates pairs in the Sports Entailment Dataset, whereas there is a small difference in the wrong direction for the ANT dataset. This helps explain the differences observed in the Base dataset precision-recall graphs A (Sports Entailment Dataset) and C (ANT dataset) in Figure 2. Furthermore, *% Overlap* has the expected correlation, showing that our method works for the temporal distribution of the sports domain data, but not for the general-domain data. That is, it can be applied successfully when there are antonymous predicate pairs that are found applying to the same argument pairs in the data, with occurrences that are temporally disjoint more often than entailing predicate pairs. In our training corpus, this distribution holds for sports predicate pairs but not for general domain predicate pairs.

Breaking down the *% Scaled* statistic per predicate pair in the ANT dataset, we do find antonyms for which many scores are scaled down, indicating that there may be predicates in the general domain where temporality is a useful signal. For example, the antonymous predicate pairs that are scaled

[9]We also include AUC scores and precision-recall plots for the Levy/Holt dataset, used in previous research (Appendix B).

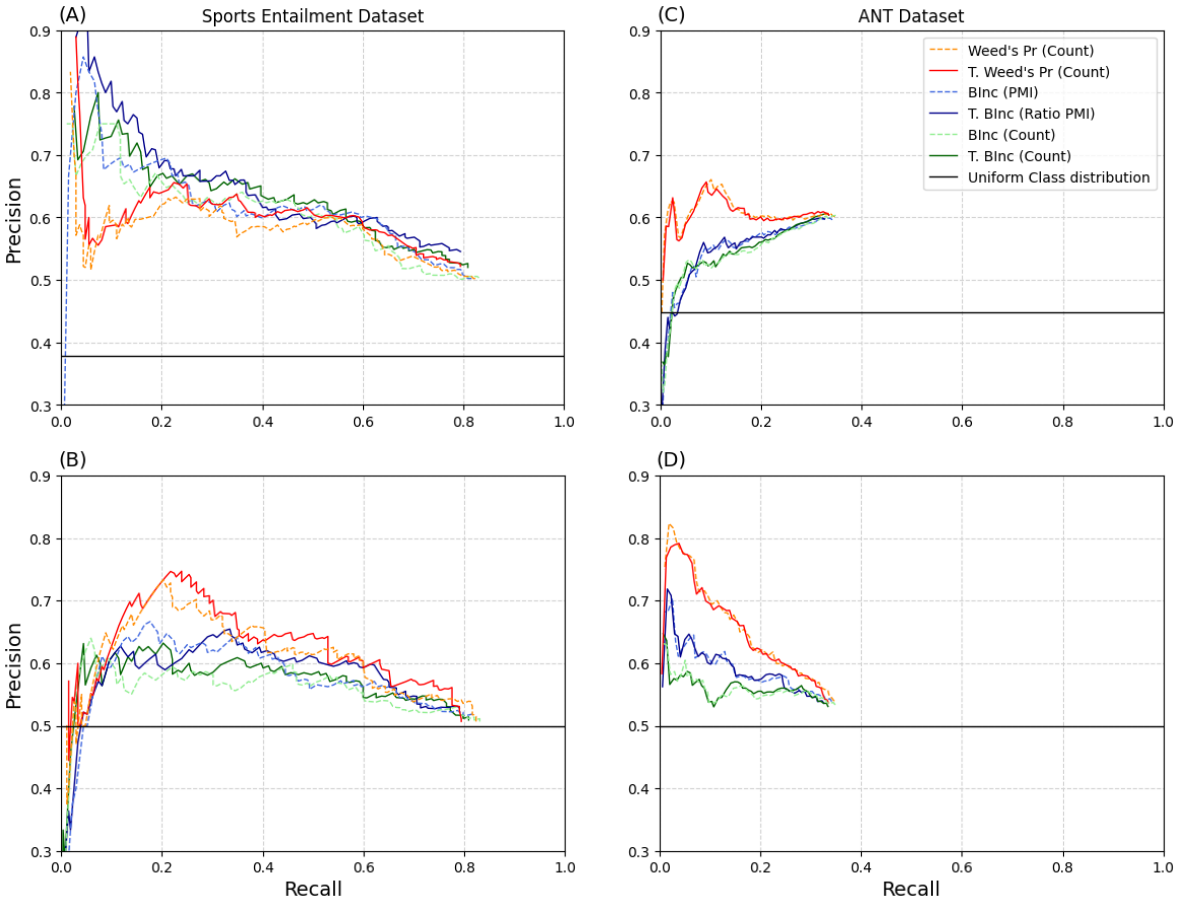

Figure 2: Precision-recall plots for the Sports Entailment Dataset (A) base and (B) directional subsets, and the ANT dataset (C) base and (D) directional subsets

most include *violate-respect*, *convict-acquit*, *allow-prohibit* and *(thing) kills (person)-(person) survives (thing)*, suggesting that predicates in legal news are worth exploring in future research. Examples found in the corpus also support this idea for other predicate pairs. We find *"Cameron, who [...], leaves London today [...]"* and *"Cameron will instead stay in London [...]"*, referring to dates a month apart. The atemporal baseline models use this data to erroneously support that *leave* ⊨ *stay in*, whereas our method successfully disentangles the evidence.

Future research could further investigate which domains or predicates stand to benefit from temporal information. This could inform models that are able to decide whether to apply temporal filtering for particular predicate pairs. Another direction is to explore how Entailment Graphs can be used to learn temporal entailments (such as *wake up - go to sleep*), which were excluded from the ANT dataset. Combining this with recent work on Multivalent Entailment Graphs will be essential here, as many of the entailment edges may be multivalent (e.g. "A kills B" ⊨ "B is dead", see also McKenna et al. (2021)). We might also consider the interaction of temporality and modality, since the temporal signal should be more able to separate antonymous data when it does not include binary relations that are stated as occurring with some degree of uncertainty (see also Guillou et al. (2021)).

8 Conclusion

We applied the temporal filtering method of Guillou et al. (2020) to the construction of Entailment Graphs for the general news domain. We evaluated the performance of the temporal filtering method on two entailment datasets. The results on the Sports Entailment Dataset suggest that a reformulation of the Distributional Inclusion Hypothesis to incorporate time would be beneficial for the sports domain. In contrast, the results on the general-domain ANT dataset suggest that the atemporal formulation of the DIH is appropriate for the general domain, although there may still be specific predicates for which the temporal formulation is effective.

References

Jonathan Berant, Noga Alon, Ido Dagan, and Jacob Goldberger. 2015. Efficient global learning of entailment graphs. *Computational Linguistics*, 41(2):221–263.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 610–619, Portland, Oregon, USA. Association for Computational Linguistics.

Sander Bijl de Vroe, Liane Guillou, Miloš Stanojević, Nick McKenna, and Mark Steedman. 2021. Modality and negation in event extraction. In *Proceedings of the 4th Workshop on Challenges and Applications of Automated Extraction of Socio-political Events from Text (CASE 2021)*, pages 31–42, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Angel X. Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3735–3740, Istanbul, Turkey. European Language Resources Association (ELRA).

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Josef Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. 1996. Using the framework. FraCaS: A framework for computational semantics.

Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34:43–69.

Maayan Geffet and Ido Dagan. 2005. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 107–114, Ann Arbor, Michigan. Association for Computational Linguistics.

Liane Guillou, Sander Bijl de Vroe, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2020. Incorporating temporal information in entailment graph mining. In *Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs)*, pages 60–71, Barcelona, Spain (Online). Association for Computational Linguistics.

Liane Guillou, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman. 2021. Blindness to modality helps entailment graph mining. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 110–116, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xavier Holt. 2018. Probabilistic models of relational implication. Master's thesis, Macquarie University.

Mohammad Javad Hosseini, Nathanael Chambers, Siva Reddy, Xavier R. Holt, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2018. Learning typed entailment graphs with global soft constraints. *Transactions of the Association for Computational Linguistics*, 6:703–717.

Mohammad Javad Hosseini, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2019. Duality of link prediction and entailment graph induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4736–4746, Florence, Italy. Association for Computational Linguistics.

Mohammad Javad Hosseini, Shay B. Cohen, Mark Johnson, and Mark Steedman. 2021. Open-domain contextual link prediction and its complementarity with entailment graphs. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2790–2802, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Thomas Kober, Sander Bijl de Vroe, and Mark Steedman. 2019. Temporal and aspectual entailment. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 103–119, Gothenburg, Sweden. Association for Computational Linguistics.

Omer Levy and Ido Dagan. 2016. Annotating relation inference in context via question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 249–255, Berlin, Germany. Association for Computational Linguistics.

Omer Levy, Ido Dagan, and Jacob Goldberger. 2014. Focused entailment graphs for open IE propositions. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 87–97, Ann Arbor, Michigan. Association for Computational Linguistics.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 94–100. AAAI Press.

Nick McKenna, Liane Guillou, Mohammad Javad Hosseini, Sander Bijl de Vroe, Mark Johnson, and Mark Steedman. 2021. Multivalent entailment graphs for question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language*

Processing, pages 10758–10768, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Merriam-Webster. 2021. Merriam-webster online thesaurus. https://www.merriam-webster.com/thesaurus. Accessed: 2021-12-16.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Saif M. Mohammad, Bonnie J. Dorr, Graeme Hirst, and Peter D. Turney. 2013. Computing lexical contrast. *Computational Linguistics*, 39(3):555–590.

Sneha Rajana, Chris Callison-Burch, Marianna Apidianaki, and Vered Shwartz. 2017. Learning antonyms with paraphrases and a morphology-aware neural network. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 12–21, Vancouver, Canada. Association for Computational Linguistics.

RelatedWords. 2021. Relatedwords.org website. https://www.relatedwords.org/. Accessed: 2021-12-16.

Enrico Santus, Frances Yung, Alessandro Lenci, and Chu-Ren Huang. 2015. EVALution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on Linked Data in Linguistics: Resources and Applications*, pages 64–69, Beijing, China. Association for Computational Linguistics.

Martin Schmitt and Hinrich Schütze. 2019. SherLIiC: A typed event-focused lexical inference benchmark for evaluating natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 902–914, Florence, Italy. Association for Computational Linguistics.

Miloš Stanojević and Mark Steedman. 2019. CCG parsing algorithm with incremental tree rotation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 228–239, Minneapolis, Minnesota. Association for Computational Linguistics.

Idan Szpektor and Ido Dagan. 2008. Learning entailment rules for unary templates. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 849–856, Manchester, UK. Coling 2008 Organizing Committee.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88.

Changlong Yu, Hongming Zhang, Yangqiu Song, Wilfred Ng, and Lifeng Shang. 2020. Enriching large-scale eventuality knowledge graph with entailment relations. In *Conference on Automated Knowledge Base Construction, AKBC 2020, Virtual, June 22-24, 2020*.

Naomi Zeichner, Jonathan Berant, and Ido Dagan. 2012. Crowdsourcing inference-rule evaluation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 156–160, Jeju Island, Korea. Association for Computational Linguistics.

Congle Zhang and Daniel S. Weld. 2013. Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1776–1786, Seattle, Washington, USA. Association for Computational Linguistics.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7579–7589, Online. Association for Computational Linguistics.

A Experimental Settings / Requirements

With the following exceptions we used MoNTEE's default settings to extract binary relations. We enabled the SUTime component to ensure that each binary relation with a predicate that could be linked to a time expression was assigned a time interval derived from SUTime [includeTemporal=True]. These time intervals were used when computing the temporal similarity measures but ignored during the computation of the non-temporal measures. We disabled unary relation extraction [writeUnaryRels=False], and restricted binary relations to only those that include at least one named entity [acceptGGBinary=False].

We used the *entGraph* framework of Hosseini et al. (2018) to construct Entailment Graphs. We raised the threshold values for infrequent predicates [minPredForArgPair=4] and argument pairs [minArgPairForPred=4] for all type-pair graphs (with the exception of the very large THING#THING graph for which we used settings of 6 and 6 respectively), and used the default values for all other parameters.

All of the experiments were conducted on a single server which has two Intel Xeon E5-2697 v4 2.3GHz CPUs (each with 18 cores) and 330GB RAM. The computational cost of training a single Entailment Graph is approximately one day and 160GB RAM. Evaluation of both the Levy/Holt and ANT datasets using the *entGraph* evaluation scripts takes approximately 6 hours per graph.

B Results on the Levy/Holt Dataset

Previous work on Entailment Graphs has reported performance on the general-domain Levy/Holt (Levy and Dagan, 2016; Holt, 2018) dataset of 18,407 entailment pairs (Hosseini et al., 2018, 2019, 2021; McKenna et al., 2021; Guillou et al., 2021). Although not designed for evaluating performance on the task of temporally separating eventualities, we also include results on the Levy/Holt dataset for the interested reader. We use the same dev/test split proposed by (Hosseini et al., 2018): 5,486 pairs for dev and 12,921 pairs for test.

AUC scores are provided in Table 3. Figure 3 contains the precision-recall plots for the Levy/Holt dev and test sets, and their directional-only components. (Note that the uniform class distribution is not shown for the dev and test sets as they fall below the 0.3 precision threshold. The uniform class distribution is 0.198 precision for dev and 0.219 precision for test.)

	Dev		Test	
Data subset	All	Dir.	All	Dir.
Recall < *threshold*	0.45	0.5	0.45	0.5
Similarity measure:				
Weed's Pr (Count)	0.215	**0.217**	0.207	**0.220**
T. Weed's Pr (Count)	0.215	0.216	0.204	0.219
BInc (PMI)	**0.221**	0.203	**0.212**	0.203
T. BInc (Ratio PMI)	**0.221**	0.199	0.209	0.203
BInc (Count)	0.217	0.208	0.205	0.201
T. BInc (Count)	0.217	0.200	0.202	0.199

Table 3: AUC scores for the Levy/Holt datasets: **All** examples and **Dir**ectional only examples for the dev and test sets. Settings: dynamic window, 5 day default.

As for the ANT dataset, we observe that performance of the temporal and non-temporal measures are very similar on the Levy/Holt dataset. This further supports the claim in Section 6 that the temporal distributions for the eventualities in the general domain are such that temporal filtering has a negligible effect.

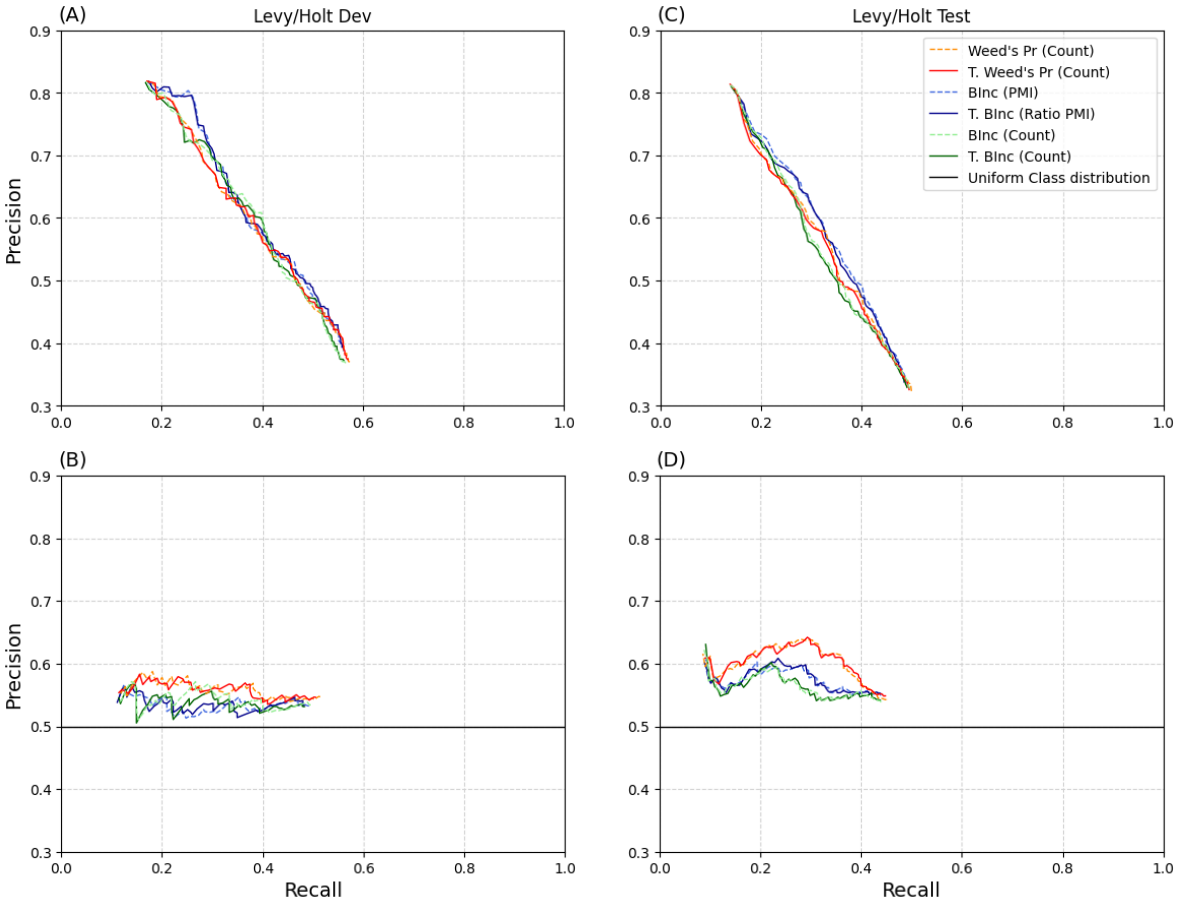

Figure 3: Precision-recall plots for the Levy/Holt dataset subsets: (A) dev, (B) dev directional-only component, (C) test, and (D) test directional-only component