Improved Induction of Narrative Chains via Cross-Document Relations

Andrew Blair-Stanek
University of Maryland
Carey School of Law
ablair-stanek@law.umaryland.edu

Benjamin Van Durme
Department of Computer Science
Johns Hopkins University
vandurme@cs.jhu.edu

Abstract

The standard approach for inducing narrative chains considers statistics gathered per individual document. We consider whether statistics gathered using cross-document relations can lead to improved chain induction. Our study is motivated by legal narratives, where cases typically cite thematically similar cases. We consider four novel variations on pointwise mutual information (PMI), each accounting for cross-document relations in a different way. One proposed PMI variation performs 58% better relative to standard PMI on recall@50 and induces qualitatively better narrative chains.

1 Introduction

Narrative chains are sets of events centered around a common protagonist. They can be induced from corpora using various unsupervised methods, many using pointwise mutual information (PMI) between events. To our knowledge, no prior work has used the information available in relations between documents in a corpus when inducing narrative chains.

To illustrate the potential for improved narrative chain induction based on document relations, we develop four novel variants of pointwise mutual information (PMI) that assume a directed graph structure between documents (i.e. relations that are edges). We test these[1] on a corpus of all U.S. federal court cases, which has a readily accessible relation between documents: citation. One case will cite prior cases as precedent in explaining its decision. We find that one of our four variants of PMI performs particularly well in the standard event cloze evaluation (Chambers and Jurafsky, 2008) and in inducing meaningful narrative chains.

2 Background

Unsupervised *narrative chain* induction from a corpus was introduced by Chambers and Jurafsky (2008), inspired by the notion of *scripts* owing to Schank and Abelson (1977). Coreference chains were extracted over the Gigaword corpus (Graff, 2002) to extract event chains with the same protagonist. A syntactic parser identified each *event* in which the protagonist was involved, defined as the combination of a verb and dependency relation, such as (convict, obj). They then calculated the pointwise mutual information (PMI) (Church and Hanks, 1989, 1990) for each combination of two events and used this PMI to do agglomerative clustering to induce narrative chains. We follow the basic approach of Chambers and Jurafsky (2008), with the major extension that we take relations between the documents into account for the first time.

There have been numerous improvements on the Chambers and Jurafsky (2008) approach, including using language modeling approaches (Rudinger et al., 2015), neural networks (Pichotta and Mooney, 2016; Weber et al., 2018), and graphs where events are the vertices (Li et al., 2018, 2021). None of these improvements has considered relations between documents in the corpus.

3 Alternative Measures

Much of the narrative chain induction literature, following Chambers and Jurafsky (2008), has used PMI. Specifically, for any given coreference chain C anywhere in the corpus, the standard PMI of two events e_1 and e_2 is defined by,

$$pmi_{standard}(e_1, e_2) = \log \frac{P(e_1 \in C \wedge e_2 \in C)}{P(e_1 \in C)P(e_2 \in C)}$$

PMI provides a measure of how often e_1 and e_2 actually occur together, as compared to what we would expect if they were independent. If they were independent, then:

$$P(e_1 \in C \wedge e_2 \in C) = P(e_1 \in C)P(e_2 \in C)$$

Note that the definition of $pmi_{standard}$ has the equation above's left hand side in the numerator and right hand side in the denominator.

[1] Code is at https://github.com/BlairStanek/cross-doc

Proceedings of the 11th Joint Conference on Lexical and Computational Semantics, pages 208 - 212

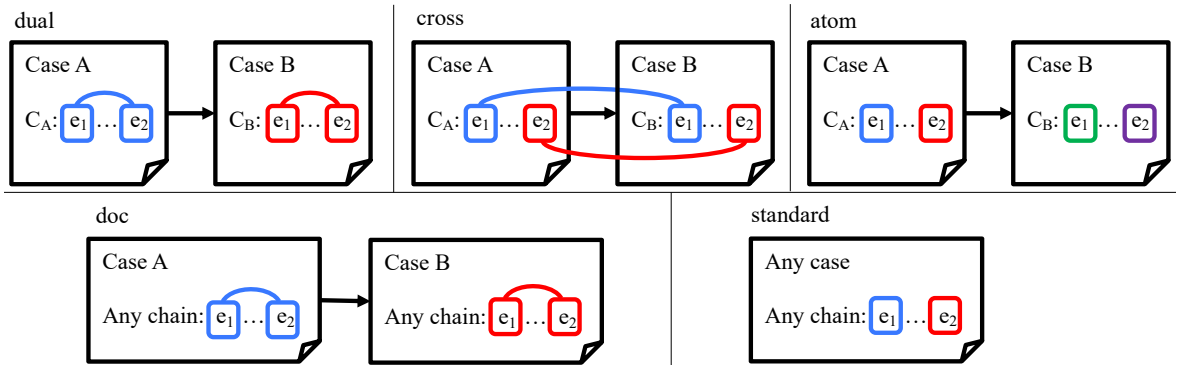

Figure 1: Illustration of events considered in the denominator of each PMI variant. Case A cites case B.

Given some relation between the documents making up the corpus, e.g. case citations, we consider four different ways to define an extension of PMI. One is document-by-document, and three are chain-by-chain.

To develop the three chain-by-chain measures, we define A and B to be documents with a relation (e.g. case A cites case B), and define C_A to be a chain of events in document A, and C_B to be a chain of events in document B. Thus, assuming independence between all occurrences of e_1 and e_2, we can derive four equivalent expressions:

$$\begin{aligned}&P(e_1 \in C_A \wedge e_2 \in C_A \wedge e_1 \in C_B \wedge e_2 \in C_B)\\ =&P(e_1 \in C_A \wedge e_2 \in C_A)P(e_1 \in C_B \wedge e_2 \in C_B)\\ =&P(e_1 \in C_A \wedge e_1 \in C_B)P(e_2 \in C_A \wedge e_2 \in C_B)\\ =&P(e_1 \in C_A)P(e_2 \in C_A)\cdot\\ &\quad P(e_1 \in C_B)P(e_2 \in C_B)\end{aligned}$$

These are the probabilities that, if you randomly select a chain C_A and a chain C_B where case A cites case B, that these chains have these events. For example, if you've randomly selected C_A and C_B, then $P(e_1 \in C_A)$ is the probability that the event e_1 appears in C_A.

By taking the first expression above as the numerator and using the last three expressions above as the denominators, we get three different extensions of PMI:

$$pmi_{dual}(e_1, e_2) = \log \frac{P(e_1, e_2 \in C_A \wedge e_1, e_2 \in C_B)}{P(e_1, e_2 \in C_A)P(e_1, e_2 \in C_B)}$$

$$pmi_{cross}(e_1, e_2) = \log \frac{P(e_1, e_2 \in C_A \wedge e_1, e_2 \in C_B)}{P(e_1 \in C_A, C_B)P(e_2 \in C_A, C_B)}$$

$$pmi_{atom}(e_1, e_2) = \log \frac{P(e_1, e_2 \in C_A \wedge e_1, e_2 \in C_B)}{P(e_1 \in C_A)P(e_2 \in C_A)P(e_1 \in C_B)P(e_2 \in C_B)}$$

A fourth approach can come from considering an analogous measure that works document-by-document, rather than chain-by-chain. Given related documents A and B (e.g. case A cites case B), if we assume that occurrences of e_1 and e_2 are independent, then the following must be true:

$$\begin{aligned}&P(\exists C_A : e_1, e_2 \in C_A \wedge \exists C_B : e_1, e_2 \in C_B)\\ =&P(\exists C_A : e_1, e_2 \in C_A)P(\exists C_B : e_1, e_2 \in C_B)\end{aligned}$$

This expression, unlike the chain-by-chain expression, cannot be further factored into two other alternatives. Why? There can be (and typically are) multiple chains in each document. Within a document, there may exist no chains with both e_1 and e_2, even though there exists a chain with e_1 and another chain with e_2.

We can get the fourth extension of PMI by dividing the two sides of the equation directly above:

$$pmi_{doc}(e_1, e_2) = \log \frac{P(\exists C_A\colon e_1, e_2 \in C_A \wedge \exists C_B\colon e_1, e_2 \in C_B)}{P(\exists C_A\colon e_1, e_2 \in C_A)P(\exists C_B\colon e_1, e_2 \in C_B)}$$

4 Experimental Setup

4.1 Dataset

We used all U.S. federal court cases since 1970 that have at least 800 total characters and that either cite to or are cited by another U.S. federal court case. All text came from the Caselaw Access Project (CAP). Cases with under 800 characters and cases neither cited to or by other federal were summary dispositions or procedural rulings that lacked meaningful description of the underlying facts of the case. The resulting corpus had 965,467 cases. (Each case is exactly one document.)

4.2 Coreference

Following Chambers and Jurafsky (2008) and subsequent literature, we extract all coreference chains from each document in the corpus. Since court decisions may be quite long (often exceeding 100,000 characters), we use the efficient long-coreference methodology of Xia et al. (2020). We hand-annotated coreference on 35 randomly selected cases (with average length of 3,518 words per case) aiming to fine-tune that model.[2] We only hand-annotated 35 cases since annotating a long document for coreference takes substantial human effort. We found that Xia et al. (2020)'s original model achieved 0.931 F1 on those 35 cases. Unfortunately, fine-tuning on splits of these 35 cases, with a variety of hyperparameters, uniformly reduced performance below this baseline.

So, we proceeded with (Xia et al., 2020)'s original model on all 965,467 cases, which took approximately 4100 hours of Quadro RTX GPU processing time. The coreference spans are available to download,[3] and we will share the spans plus tokens with those with researcher approval from the Caselaw Access Project.

4.3 Parsing and Chain Extraction

We use Stanford CoreNLP (Manning et al., 2020) for syntactic parsing, including POS tagging, lemmatization, and dependency parsing. We then use PredPatt (White et al., 2016) to extract predicates and arguments from the dependency parse. If an argument matches one of the entities identified during coreference, we consider the event as a 2-tuple of the predicate's lemma and the dependency type (e.g. (convict, obj)). Although the predicate is often a verb, it need not be, unlike in Chambers and Jurafsky (2008), which restricted predicates to being verbs. We retained all chains of length 2 or more; most cases had multiple chains. We do not follow Chambers and Jurafsky (2008) in attempting partial temporal ordering of events. Thus, each chain is an unordered set of events that shares the same co-referring entity.

Using these chains, we calculated all four of our proposed PMI variations that rely on the relations between documents (i.e., citations between cases). We also calculated $pmi_{standard}$, which does not rely on the relations. All these training calculations were by CPU and ran on the entire corpus, except for some cases held out for testing. So, the calculations were run on 955,810 cases, between which there were 10,606,964 citation relations, containing a total of 27,166,457 chains and 24,364,877,760 combinations of chain C_A from case A and chain C_B from case B, where case A cites case B. The complete set of event chains from each case are available for download.[4]

Measure	R@1	R@5	R@50	MRR
$\text{pmi}_{standard}$	1.7%	4.9%	15.9%	0.037
pmi_{dual}	0.4%	1.2%	6.1%	0.011
pmi_{cross}	**2.1%**	**6.6%**	**25.2%**	**0.050**
pmi_{atom}	1.4%	4.5%	19.0%	0.036
pmi_{doc}	0.4%	1.2%	6.3%	0.012

Table 1: Cloze Performance on test set of 27,324 held-out chains, measured by Recall@1, Recall@5, Recall@50, and Mean Reciprocal Rank.

5 Results and Discussion

5.1 Quantitative evaluation

We measure the effectiveness of the different measures of PMI using the event cloze task, following Chambers and Jurafsky (2008), where we randomly remove an event from each test chain and use the PMI measures to predict what event should fill that. For test, we used 0.3 percent of the corpus (2783 cases) that had been held back and not used to calculate any of the PMI measures, either as a citing case or cited case. We used all chains with 3 or more events, which resulted in 27,324 chains used for the cloze test, each with one event randomly removed. (We used chains with 3 or more events because, when removing one event for cloze, that leaves chains with 2 or more events.) Each possible event that might complete the cloze is evaluated as the sum of the PMIs with the other events in the chain (i.e. other than the one removed). We measure performance in several ways: the percentage of chains where the correct event is the top prediction (recall@1); within the top 5 predictions (recall@5); within the top 50 predictions (recall@50); and, finally, mean reciprocal rank (MRR).

Looking at Table 1, we see that two of our four PMI variants substantially underperform standard PMI: pmi_{doc} and pmi_{dual}. It is worth noting that the former is just a document-by-document version

[2] The full 35 annotations are at https://doi.org/10.7281/T1/QVAHMD

[3] https://doi.org/10.7281/T1/QVAHMD

[4] https://doi.org/10.7281/T1/QVAHMD

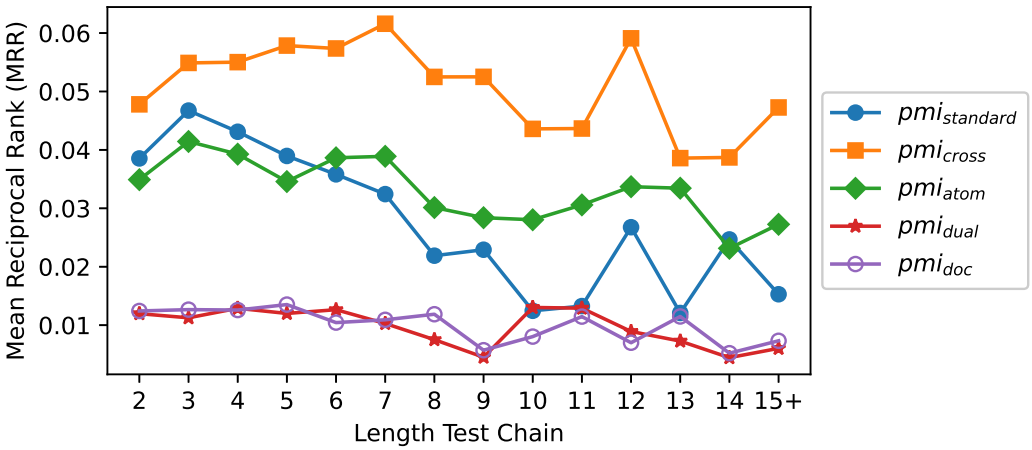

Figure 2: Performance of the five types of PMI, measured by mean reciprocal rank, by length of the test chain. For example, a test chain of length 2 originally had 3 events, one of which is removed for cloze prediction, and the reported performance is how well that PMI measure predicts the actual removed event.

of the latter. Both compare the frequency of both events e_1 and e_2 in both a cited and citing case to the frequency of both events in cases by themselves. We hypothesize that the decision of a court to cite a previous decision is noisy and unpredictable, so that even when both an earlier case and a later case have a chain with both e_1 and e_2, the decision of the judge authoring the later case to cite the earlier case is noisy.

By contrast, pmi_{cross} normalizes out the noisiness of the decision of whether to cite or not. Its denominator uses the probability that e_1 is in both C_A and C_B multiplied by the probability that e_2 is in both C_A and C_B. By definition, these probabilities already take into account the decision of the author of the later case A to cite the earlier case B. We observe that pmi_{cross} substantially outperforms the standard $pmi_{standard}$ that has been the foundation for most narrative chain induction work, achieving a recall@50 of 25.2% (versus 15.9%, a 58% relative improvement) and a mean reciprocal rank (MRR) of 0.050 (versus 0.037).

Note that the cloze test used for this evaluation runs entirely on chains within a single case, not relying in any way on citation relations between cases. Yet our newly introduced pmi_{cross}, which is calculated using the citation relations, outperforms $pmi_{standard}$, which is calculated solely on chains within single cases and does not use the relations.

To determine whether these trends in performance are attributable to chains of a particular length, in Figure 2 we graph all five variations of PMI by chain length. We see that pmi_{cross} outperforms all other measures, including $pmi_{standard}$ for all chain lengths.

5.2 Qualitative evaluation

High-quality narrative chains should correspond to sensible groupings of events actually encountered. A U.S.-trained lawyer reviewed a sample of chains from both and found that the narrative chains induced using pmi_{cross} and agglomerative clustering are qualitatively better than those induced in the same way but using $pmi_{standard}$. To do agglomerative clustering, we build a cluster around every set of two events that appears in any chain, and we repeatedly add the event with the highest sum of PMIs with the existing events, until we reach a desired maximum set size (we used 6). These sets are the narrative chains. We use dynamic programming to avoid duplication, and we rank the final clusters by the total sum of PMIs between all elements. Here are two 6-event-long example narrative chains induced using pmi_{cross} that were not induced using $pmi_{standard}$. One relates to a criminal defendant and the other relates to a trademark being found generic and thus invalid (as happened to Kleenex's trademark):

(have, nsubj)	(trademark, nsubj)
(commit, nsubj)	(mark, nsubj)
(perpetrate, nsubj)	(term, nsubj)
(plead, nsubj)	(use, nsubj:pass)
(sentence, obj)	(descriptive, nsubj)
(serve, nsubj)	(generic, nsubj)

6 Conclusion

We have explored four new measures of PMI that can take advantage of relationships between documents in corpora. Applying them to the corpus of federal cases, we find that one such measure, pmi_{cross} shows substantial improvement over standard PMI. Future work may consider the use of these new PMI measures on other corpora where the documents may have relationships that can be characterized as directed edges, including hyperlinks and references.[5]

We focused on a PMI-based approach to inducing narrative chains owing to its familiarity within the community. Based on these results demonstrating the benefits of utilizing document-to-document relations, future work can consider extensions such as using temporal relations, causality, and neural modeling.

Acknowledgements

The authors would like to thank Patrick Xia, Nils Holzenberger, Rachel Rudinger, Anton Belyy, Noah Weber, and the anonymous reviewers.

References

Caselaw access project. https://case.law. Accessed: 2020.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of the Association of Computational Linguistics*, pages 789–797.

Kenneth Church and Patrick Hanks. 1989. Word association norms, mutual information, and lexicography. In *ACL 27th Annual Meeting*, pages 76–83.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

David Graff. 2002. English gigaword. *Linguistic Data Consortium*.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Constructing narrative event evolutionary graph for script event prediction. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, pages 4201–4207.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2020. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Karl Pichotta and Raymond J. Mooney. 2016. Learning statistical scripts with LSTM recurrent neural networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2800–2806.

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script induction as language modeling. In *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Hierarchical quantized representations for script generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3783–3792.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723.

Patrick Xia, João Sedoc, and Benjamin Van Durme. 2020. Incremental neural coreference resolution in constant memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8617–8624.

[5] Our new PMI measures can trivially be extended to relationships that are undirected edges or even weighted edges.