Listwise temporal ordering of events in clinical notes

Serena Jeblee Department of Computer Science University of Toronto Toronto, Ontario, Canada sjeblee@cs.toronto.edu

Abstract

We present metrics for listwise temporal ordering of events in clinical notes, as well as a baseline listwise temporal ranking model that generates a timeline of events that can be used in downstream medical natural language processing tasks.

1 Introduction

For medical narratives such as clinical notes, event and time information can be useful in automated classification and prediction tasks. For example, the timeline of a patient's medical history can be used to predict whether they will be readmitted to the hospital within a certain time window. A medical timeline can also be used for other tasks such as disease classification, and for summarizing a patient history for physicians.

Because events are not necessarily mentioned in chronological order in such documents, once the individual events are identified, the model needs to determine the temporal relationships between them. Temporal relations are categorical labels that describe how two events are related. These relations can be binary (related or not) or simple (BEFORE, AFTER, OVERLAP), or they can capture more complex relationships such as partial overlap or adjacency. A popular temporal relation scheme for clinical notes is the CONTAINS relation, which specifies whether a time phrase or event subsumes another event.

However, most temporal relation methods use pairwise classification, which can result in inconsistent relationships and which requires classifying n^2 pairs of events, many of which have no defined relation. What is needed is an overall timeline of medically relevant events that ideally can capture event duration and overlap. A listwise ordering of events inherently captures all pairwise Graeme Hirst Department of Computer Science University of Toronto Toronto, Ontario, Canada gh@cs.toronto.edu

relationships between events and prevents inconsistencies that can arise in pairwise ordering.

While ranking methods have generally been applied to information retrieval tasks such as searching, we can view temporal ordering as a ranking task. In this work, we examine a baseline listwise ranking method for events in clinical notes and we establish a set of metrics for evaluating listwise temporal ordering of these events.

1.1 Listwise vs. pairwise ordering

Temporal relation extraction is typically framed as a pairwise classification problem: generate all pairs of events in a document, and then determine what type of temporal relation exists between them, if any. The major problem with this approach is that the vast majority of event pairs have no relationship, or the relation between them is unknown. This results in an unbalanced classification problem, and there is no guarantee that the predicted pairwise relations are consistent with one another. Because of the sparsity of annotated long-distance relations, many pairwise classification models have been limited to events mentioned within the same sentence or within some small window of the text. It is often difficult for humans to analyze the relations from an entire document quickly, especially when they are inconsistent.

In contrast, a document-level list inherently captures pairwise relations between all events in the document, regardless of whether or not they appear in the same sentence. Thus, we choose to represent the events as a temporally ordered list instead of as pairs of temporal relations.

However, since pairwise relations often capture relationships that are more complex than just BE-FORE, AFTER, and OVERLAP, we add time information to the events in the reference list when available. This information includes event start, end, and overlap times, based on the annotated relationships to time phrases in the text. For this work, we sort the list by event start time, but in principle we could sort by end time or examine event overlaps. All time information can be either exact, relative (before or after a certain time), or unknown.

2 Related work

Most existing work on temporal relation extraction for clinical text relies on human-labeled spans and relations. The input to these models is usually pairs of events (or events and times) that a human has identified as being related, and all the model has to do is decide the type of relation. However, given an unlabeled dataset the task is much more difficult – the system must first identify the events and time phrases, decide which pairs are related, and then determine the type of each relation. Derczynski (2017) covers the general topic of temporal ordering of events in text.

For the medical domain, the Clinical TempEval task at SemEval (Bethard et al., 2017) has multiple tasks that involve identification of events, time expressions, and attributes in clinical notes, as well as relation classification. SemEval 2015 also had a task on cross-document event ordering, although the data was in the news domain (Minard et al., 2015).

Additionally, most recent work has focused on small relation sets, such as narrative container relations (CONTAINS, NO-RELATION), which were originally introduced by Pustejovsky and Stubbs (2011), or simple relations (BEFORE, AFTER, OVERLAP, NONE), although some work has attempted to classify with Allen's complete set of 13 temporal relations (Allen, 1984).

Dligach et al. (2017) and Lin et al. (2017) achieved state-of-the-art performance on identifying container relations in the THYME corpus (Styler et al., 2014); however, they considered only relations in which both entities appear in the same sentence. This is a limitation in many temporal relation systems. Since clinical notes are often long and may refer to distant entities such as the admission or discharge date, crosssentence relations should not be ignored. Tourille et al. (2017) identified cross-sentence container relations in the THYME corpus, in addition to intrasentence relations, using a bi-directional LSTM. They used word and character embeddings of gold-standard event attributes and attributes generated by cTAKES (Savova et al., 2010).

Tannier and Muller (2011) addressed relation closure in temporal graphs with all 13 Allen relations. In our current work we deal only with simple relations, but this is something we would like to expand in the future.

For our ranking model, we build upon List-Net (Cao et al., 2007), which describes a listwise approach to ranking. The ranking function is a linear neural network which assigns a relevance score for each document in a set related to a query (such as in a document retrieval task). The loss function is typically based on top-k probability, i.e., the probability of a given document being ranked among the top k documents with respect to a query. More recent work such as IntervalRank (Moon et al., 2010) used isotonic regression with a maximum margin criteria to optimize for correct relative rankings.

3 Data

3.1 Dataset

We use the THYME corpus (Styler et al., 2014), which contains de-identified clinical notes with human-annotated times, events, and temporal relations, using the TimeML schema (Pustejovsky et al., 2003). This dataset is publicly available with a data use agreement. We use the provided train/dev/test split and the gold-standard EVENT, TIMEX3, and temporal relation annotations, including document creation time (DCT) relations.

For now, our listwise ordering method can represent only simple relations (BEFORE, AFTER, OVERLAP), so we map the BEFORE/OVERLAP relation and ENDS-ON relation to BEFORE (since we are ranking by start time), we transform AFTER relations to BEFORE, and all other relations in the THYME dataset to OVERLAP (including the CON-TAINS relation).

One of the limitations of the annotations in the THYME dataset is that event annotations are always applied to just a single word, even though there are many instances where the event would be better represented by a phrase. Unfortunately, this is common in temporally annotated datasets.

3.2 Converting gold-standard pairwise relations to list representations

In order to evaluate listwise ordering methods, we need a reference list to compare against. To our knowledge, all temporally annotated clinical



Figure 1: Example of the type of cycle in the temporal graph where the OVERLAP link would be dropped.

datasets have pairwise annotations, so we convert these pairwise relations into reference lists for training our model. We use the graph of gold-standard pairwise event relations to extract a grouped listwise ordering. This is not a straightforward process, since not all event pairs are annotated. The vast majority of relations are between event and the document creation time (DCT), which makes it difficult to determine how events are related to each other when there is no explicit annotation for that pair.

First, we take only the gold-standard eventevent relations and create a directed graph representation¹. Unfortunately, we find that some of these graphs have cycles, which indicate inconsistent orderings in the gold-standard annotation (such as A BEFORE B, B BEFORE C, C OVER-LAP A). We have no choice but to drop some relation links in order to resolve these cycles. We choose to drop OVERLAP links, since these are the least specific, as the relation it does not specify how the events overlap. Since we are ordering by start time, for two events to have different ranks means only that one starts before the other; it doesn't mean that they don't overlap. Therefore we favor preserving the BEFORE relations. In total, 30 OVERLAP links were removed from the test set. See Figure 1 for a fabricated example of the type of inconsistency where the OVERLAP link is dropped.

We then augment the graph with transitive and time-based relations. For annotated event-time relations, we add the associated time information to the event, along with the part of the interval that the time specifies (start, end, or overlap). We use the Python *dateparser* module to convert the string representation to an ISO date-time format. We then compare the time intervals of every pair of events to discover more BEFORE and AFTER relations. We compare the start and end times of events first, and if that information is not available, we compare the overlap times of the two events.

Lastly, we group the events that all have the same incoming and outgoing relations and have either overlap relations or no specified relations with each other. This results in a number of 'bins', which can each contain one or more events, and all of the relations from the individual events. We then order these bins according to the BEFORE and AFTER relations between bins, which are preserved from the individual events. All events in the same bin are assigned the same rank. The final list of events, including associated time information, can be easily viewed and understood by humans.

We verify that the output list preserves the pairwise relations by checking that for each eventevent relation in the original set, the events are ordered correctly in the list. For time-event relations, we check that the associated interval information is consistent with the time relation. As discussed above, we are forced to ignore some of the OVERLAP links and leave the events in separate bins because combining them would create conflicts between the merged edges. We also note that there may be many variants of the listwise ordering that are consistent with the pairwise goldstandard relations.

The list conversion code is available at https: //github.com/sjeblee/chrononet.

4 Listwise evaluation metrics

Traditional ranking models are usually evaluated according to normalized discounted cumulative gain (NDCG) and mean average precision (MAP). However, both of these metrics are focused on the top k ranked documents, which makes sense for a document retrieval task but is not an appropriate metric for temporal ranking, where we care about the ordering of all events.

Here we present two listwise ranking metrics, in addition to the standard pairwise recall:

Mean squared error (MSE)

$$MSE = \frac{\sum_{y \in Y} (rank_t(y) - rank_p(y))^2}{|Y|} \qquad (1)$$

where Y is the set of events, $rank_t$ is the correct rank, and $rank_p$ is the predicted rank. This is an absolute metric that measures how correct the rank score is for each individual event. However, this

¹AFTER relations are inverted to become BEFORE relations, and INITIALIZES and FINISHES are converted to BEGINS-ON and ENDS-ON respectively.

does not measure how correct the relative rankings are, so we introduce a second metric:

Pairwise ordering accuracy (POA)

$$POA = \frac{p_O + p_E}{|L_O| + |L_E|} \tag{2}$$

$$p_{O} = \sum_{u,v \in L_{O}} \begin{cases} 1, & \text{if } rank_{p}(u) < rank_{p}(v) \\ 0, & \text{otherwise} \end{cases}$$
(3)

$$p_{E} = \sum_{u,v \in L_{E}} \begin{cases} 1, & \text{if } |rank_{p}(u) - rank_{p}(v)| \le \varepsilon \\ 0, & \text{otherwise} \end{cases}$$
(4)

where L_O is the set of ordered pairs (u, v) in the reference list such that $rank_t(u) < rank_t(v)$ (i.e., event *u* is ranked before event *v*), and L_E is the set of pairs (u, v) where $rank_t(u) = rank_t(v)$.

Since the ranking model may output slightly different rank values for events that are close together, we consider rank values to be the same if they are within ε of each other. For this paper we set $\varepsilon = 0.01$, and rank values are between 0 and 1 inclusive, with 0 being the earliest start time.

Although this metric looks at pairs of events, it measures the overall accuracy of the whole list. If two events are swapped but are otherwise in roughly the correct position in the list, POA will penalize the model less than it would for an event that is placed far away from its correct position.

Gold-standard pairwise relation recall (GPR)

From the list output, it is easy to extract all event– event relation pairs. From these we can compute the pairwise classification accuracy. Since many event–event relations are not present in the goldstandard annotations, we report recall only.

5 Models

For our ranking model we use an open-source implementation of ListNet², substituting rank MSE as the loss function.

The model input is the concatenation of an embedding vector and a normalized vector of numerical features. The embedding vector contains the word embedding of the event text concatenated with the word embeddings of the previous and next 3 words, and the second feature vector contains the gold-standard event attributes and the

Model	MSE	POA	GPR _B	GPR _O
Reference list	.000	1.000	.844	.246
ListNet	.072	.517	.420	.254
Text order	.148	.413	.640	.000
Random ordering	.170	.366	.485	.000
Pairwise NN	_	_	.186	.624

Table 1: Listwise ordering on the THYME test set. MSE: mean squared error, POA: list pairwise ordering accuracy, GPR_B : gold-standard pairwise relation recall of BEFORE relations, GPR_O : overlap relations

Model	Р	R	F ₁
Pairwise NN	.006	.540	.011
Pairwise NN (no NONE)	.841	.851	.825

Table 2: Pairwise relation classification on the THYME test set. The first pairwise neural network (NN) model includes all possible pairs, including NONE relations. The second model is restricted to only pairs that are known to have a relation. P: precision, R: recall

span start of the event. We use publicly available word embeddings trained on Wikipedia, PubMed, and PMC (Pyysalo et al., 2013). The target rank of each event is the position in the reference list, scaled to [0,1]. Any number of events can share the same rank.

The pairwise classification model is a feedforward neural network implemented in PyTorch (Paszke et al., 2017), with one hidden layer with 256 nodes and ReLU activations, trained for 10 epochs. Each event pair is represented with the same features as the ranking model, and the scaled character distance between the two events in the text. The goal of this classification model is not to beat the state of the art, but rather to compare the listwise method to a simple pairwise model.

6 Results

Table 1 shows the accuracy of the ListNet ranking model according to the listwise metrics (MSE and POA), as well as gold-standard pair relation recall (GPR). As a baseline, we include the results from random ranking (every event is randomly assigned a ranking value between 0 and 1), and ranking by the order of mention in the text (since many events are indeed mentioned in chronological order). Scores from random ranking are averaged over 10 runs. We also include GPR results from the pairwise classification model for comparison.

²https://github.com/shiba24/ learning2rank

While the ListNet ranking model has plenty of room for improvement in terms relative ordering, it outperforms both the random ordering and the order of mention in the text.

Table 2 shows the accuracy of the pairwise classification with respect to the gold-standard annotations. We cannot extract a listwise ordering from the pairwise model results because the predicted relations have cycles. In addition, most temporal relation models using THYME data have used the full set of relations or only container relations, and thus are not comparable to this model.

7 Discussion and future work

For many health-related NLP tasks, listwise ordering offers several benefits over pairwise ordering. The list avoids cycles and inconsistent pair relations, and is also a more compact representation – all pairwise relations can be inferred from the list. Moreover, the list of events and associated time information is easy for humans to review.

Although simple listwise ordering does not capture more-finely grained interval temporal relations such as partial event overlap and endpoint relations, the inclusion of interval time information for each event allows us to choose how to order them. For example, we could choose to order the list by end time instead of start time. In the future we hope to represent more-complex event relations and handle relative time phrases.

8 Conclusion

We have shown that events in clinical text can be ordered in a listwise fashion, which prevents many of the issues that occur in pairwise classification. The metrics presented here are an alternative to pairwise-only metrics, which we hope will serve as a foundation for further listwise temporal ordering work in the medical domain.

Acknowledgments

The authors thank Prabhat Jha and Mireille Gomes for their advice, and the Mayo Clinic for providing the THYME dataset through the Health Natural Language Processing Center. This work was supported by a Google Faculty Award, the U.S. National Institutes of Health, and the University of Toronto.

References

- James F Allen. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.
- Steven Bethard, Guergana Savova, Martha Palmer, and James Pustejovksy. 2017. SemEval-2017 Task 12: Clinical TempEval. In Proceedings of the 11th Workshop on Semantic Evaluations (SemEval 2017), pages 565–572. Association for Computational Linguistics.
- Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the* 24th International Conference on Machine Learning, ICML '07, pages 129–136, New York, NY, USA. ACM.
- Leon RA Derczynski. 2017. Automatically Ordering Events and Times in Text, volume 677 of Studies in Computational Intelligence. Springer International Publishing.
- Dmitriy Dligach, Timothy Miller, Chen Lin, Steven Bethard, and Guergana Savova. 2017. Neural temporal relation extraction. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 746–751. Association for Computational Linguistics.
- Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2017. Representations of time expressions for temporal relation extraction with convolutional neural networks. In *Proceedings of the 16th Workshop on Biomedical Natural Language Processing*, pages 322–327. Association for Computational Linguistics.
- Anne-Lyse Minard, Manuela Speranza, Eneko Agirre, Itziar Aldabe, Marieke Van Erp, Bernardo Magnini, German Rigau, and Fondazione Bruno Kessler. 2015. SemEval-2015 Task 4: TimeLine: Crossdocument event ordering. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), (SemEval):778–786.
- Taesup Moon, Alex Smola, Yi Chang, and Zhaohui Zheng. 2010. IntervalRank: Isotonic regression with listwise and pairwise constraints. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 151–160, New York, NY, USA. ACM.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In NIPS Workshop.
- James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. In *IWCS-5*,

Fifth International Workshop on Computational Semantics, pages 1–11.

- James Pustejovsky and Amber Stubbs. 2011. Increasing informativeness in temporal annotation. In *Proceedings of the 5th Linguistic Annotation Workshop*, pages 152–160.
- Sampo Pyysalo, Filip Ginter, Hans Moen, Tapio Salakoski, and Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In *Proceedings of the Fifth International Symposium on Languages in Biology and Medicine*, pages 39–44.
- Guergana K Savova, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute1. 2010. Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications. *Journal of the American Medical Informatics Association: JAMIA*, 17(5):507–513.
- William F Styler, IV, Steven Bethard, Sean Finan, Martha Palmer, Sameer Pradhan, Piet C de Groen, Brad Erickson, Timothy Miller, Chen Lin, Guergana Savova, and James Pustejovsky. 2014. Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2014(2):143–154.
- Xavier Tannier and Philippe Muller. 2011. Evaluating temporal graphs built from texts via transitive reduction. *Journal of Artificial Intelligence Research*, 40:375–413.
- Julien Tourille, Olivier Ferret, Aurélie Névéol, and Xavier Tannier. 2017. Neural architecture for temporal relation extraction: A bi-LSTM approach for detecting narrative containers. In *Proceedings of the* 55th Annual Meeting of the Association for Computational Linguistics, pages 224–230.