Inducing Temporal Graphs

Philip Bramsen MIT CSAIL bramsen@mit.edu

Pawan Deshpande MIT CSAIL pawan@mit.edu

Yoong Keok Lee DSO National Laboratories lyoongke@dso.org.sg

Regina Barzilay MIT CSAIL regina@csail.mit.edu

Abstract

We consider the problem of constructing a directed acyclic graph that encodes temporal relations found in a text. The unit of our analysis is a temporal segment, a fragment of text that maintains temporal coherence. The strength of our approach lies in its ability to simultaneously optimize pairwise ordering preferences and global constraints on the graph topology. Our learning method achieves 83% F-measure in temporal segmentation and 84% accuracy in inferring temporal relations between two segments.

1 Introduction

Understanding the temporal flow of discourse is a significant aspect of text comprehension. Consequently, temporal analysis has been a focus of linguistic research for quite some time. Temporal interpretation encompasses levels ranging from the syntactic to the lexico-semantic (Reichenbach, 1947; Moens and Steedman, 1987) and includes the characterization of temporal discourse in terms of rhetorical structure and pragmatic relations (Dowty, 1986; Webber, 1987; Passonneau, 1988; Lascarides and Asher, 1993).

Besides its linguistic significance, temporal analysis has important practical implications. In multidocument summarization, knowledge about the temporal order of events can enhance both the content selection and the summary generation processes (Barzilay et al., 2002). In question answering, temporal analysis is needed to determine when a particular event occurs and how events relate to each other. Some of these needs can be addressed by emerging technologies for temporal analysis (Wilson et al., 2001; Mani et al., 2003; Lapata and Lascarides, 2004; Boguraev and Ando, 2005).

This paper characterizes the temporal flow of discourse in terms of *temporal segments* and their ordering. We define a temporal segment to be a fragment of text that does not exhibit abrupt changes in temporal focus (Webber, 1988). A segment may contain more than one event or state, but the key requirement is that its elements maintain temporal coherence. For instance, a medical case summary may contain segments describing a patient's admission, his previous hospital visit, and the onset of his original symptoms. Each of these segments corresponds to a different time frame, and is clearly delineated as such in a text.

Our ultimate goal is to automatically construct a graph that encodes ordering between temporal segments. The key premise is that in a coherent document, temporal progression is reflected in a wide range of linguistic features and contextual dependencies. In some cases, clues to segment ordering are embedded in the segments themselves. For instance, given a pair of adjacent segments, the temporal adverb *next day* in the second segment is a strong predictor of a precedence relation. In other cases, we can predict the right order between a pair of segments by analyzing their relation to other segments in the text. The interaction between pairwise ordering decisions can easily be formalized in terms of constraints on the graph topology. An obvious example of such a constraint is prohibiting cycles in the ordering graph. We show how these complementary sources of information can be incorporated in a model using global inference.

We evaluate our temporal ordering algorithm on a corpus of medical case summaries. Temporal analysis in this domain is challenging in several respects: a typical summary exhibits no significant tense or aspect variations and contains few absolute time markers. We demonstrate that humans can reliably mark temporal segments and determine segment ordering in this domain. Our learning method achieves 83% F-measure in temporal segmentation and 84% accuracy in inferring temporal relations between two segments.

Our contributions are twofold:

Temporal Segmentation We propose a fully automatic, linguistically rich model for temporal segmentation. Most work on temporal analysis is done on a finer granularity than proposed here. Our results show that the coarse granularity of our representation facilitates temporal analysis and is especially suitable for domains with sparse temporal anchors.

Segment Ordering We introduce a new method for learning temporal ordering. In contrast to existing methods that focus on pairwise ordering, we explore strategies for global temporal inference. The strength of the proposed model lies in its ability to simultaneously optimize pairwise ordering preferences and global constraints on graph topology. While the algorithm has been applied at the segment level, it can be used with other temporal annotation schemes.

2 Related Work

Temporal ordering has been extensively studied in computational linguistics (Passonneau, 1988; Webber, 1988; Hwang and Schubert, 1992; Lascarides and Asher, 1993; Lascarides and Oberlander, 1993). Prior research has investigated a variety of language mechanisms and knowledge sources that guide interpretation of temporal ordering, including tense, aspect, temporal adverbials, rhetorical relations and pragmatic constraints. In recent years, the availability of annotated corpora, such as TimeBank (Pustejovsky et al., 2003), has triggered the use of machine-learning methods for temporal analysis (Mani et al., 2003; Lapata and Lascarides, 2004; Boguraev and Ando, 2005). Typical tasks include identification of temporal anchors, linking events to times, and temporal ordering of events.

Since this paper addresses temporal ordering, we focus our discussion on this task. Existing ordering approaches vary both in terms of the ordering unit — it can be a clause, a sentence or an event — and in terms of the set of ordering relations considered by the algorithm. Despite these differences, most existing methods have the same basic design: each pair of ordering units (i.e., clauses) is abstracted into a feature vector and a supervised classifier is employed to learn the mapping between feature vectors and their labels. Features used in classification include aspect, modality, event class, and lexical representation. It is important to note that the classification for each pair is performed independently and is not guaranteed to yield a globally consistent order.

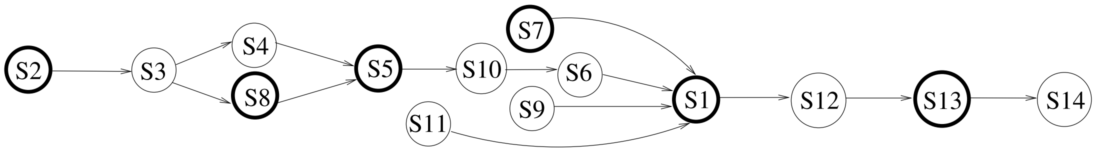
In contrast, our focus is on globally optimal temporal inference. While the importance of global constraints has been previously validated in symbolic systems for temporal analysis (Fikes et al., 2003; Zhou et al., 2005), existing corpus-based approaches operate at the local level. These improvements achieved by a global model motivate its use as an alternative to existing pairwise methods.

3 TDAG: A representation of temporal flow

We view text as a linear sequence of temporal segments. Temporal focus is retained within a segment, but radically changes between segments. The length of a segment can range from a single clause to a sequence of adjacent sentences. Figure 1 shows a sample of temporal segments from a medical case summary. Consider as an example the segment S13 of this text. This segment describes an examination of a patient, encompassing several events and states (i.e., an abdominal and neurological examination). All of them belong to the same time frame, and temporal order between these events is not explicitly outlined in the text.

We represent ordering of events as a temporal directed acyclic graph (TDAG). An example of the transitive reduction[1] of a TDAG is shown in Figure 1. Edges in a TDAG capture temporal precedence relations between segments. Because the graph encodes an order, cycles are prohibited. We do not require the graph to be fully connected — if the precedence relation between two nodes is not specified in the text, the corresponding nodes will not be connected. For instance, consider the segments S5 and S7 from Figure 1, which describe her previous tests and the history of eczema. Any

[1] The transitive reduction of a graph is the smallest graph with the same transitive closure.



S1	A 32-year-old woman was admitted to the hospital because of left subcostal pain...
S2	The patient had been well until four years earlier,
S5	Three months before admission an evaluation elsewhere included an ultrasonographic examination, a computed tomographic (CT) scan of the abdomen...
S7	She had a history of eczema and of asthma...
S8	She had lost 18 kg in weight during the preceding 18 months.
S13	On examination the patient was slim and appeared well. An abdominal examination revealed a soft systolic bruit... and a neurologic examination was normal...

Figure 1: An example of the transitive reduction of a TDAG for a case summary. A sample of segments corresponding to the nodes marked in bold is shown in the table.

order between the two events is consistent with our interpretation of the text, therefore we cannot determine the precedence relation between the segments S5 and S7.

In contrast to many existing temporal representations (Allen, 1984; Pustejovsky et al., 2003), TDAG is a coarse annotation scheme: it does not capture interval overlap and distinguishes only a subset of commonly used ordering relations. Our choice of this representation, however, is not arbitrary. The selected relations are shown to be useful in text processing applications (Zhou et al., 2005) and can be reliably recognized by humans. Moreover, the distribution of event ordering links under a more refined annotation scheme, such as TimeML, shows that our subset of relations covers a majority of annotated links (Pustejovsky et al., 2003).

4 Method for Temporal Segmentation

Our first goal is to automatically predict shifts in temporal focus that are indicative of segment boundaries. Linguistic studies show that speakers and writers employ a wide range of language devices to signal change in temporal discourse (Bestgen and Vonk, 1995). For instance, the presence of the temporal anchor *last year* indicates the lack of temporal continuity between the current and the previous sentence. However, many of these predictors are heavily context-dependent and, thus, cannot be considered independently. Instead of manually crafting complex rules controlling feature interaction, we opt to learn them from data.

We model temporal segmentation as a binary classification task. Given a set of candidate boundaries (e.g., sentence boundaries), our task is to select a subset of the boundaries that delineate temporal segment transitions. To implement this approach, we first identify a set of potential boundaries. Our analysis of the manually-annotated corpus reveals that boundaries can occur not only between sentences, but also within a sentence, at the boundary of syntactic clauses. We automatically segment sentences into clauses using a robust statistical parser (Charniak, 2000). Next, we encode each boundary as a vector of features. Given a set of annotated examples, we train a classifier[2] to predict boundaries based on the following feature set:

Lexical Features Temporal expressions, such as *tomorrow* and *earlier*, are among the strongest markers of temporal discontinuity (Passonneau, 1988; Bestgen and Vonk, 1995). In addition to a well-studied set of domain-independent temporal markers, there are a variety of domain-specific temporal markers. For instance, the phrase *initial hospital visit* functions as a time anchor in the medical domain.

To automatically extract these expressions, we provide a classifier with n-grams from each of the candidate sentences preceding and following the candidate segment boundary.

Topical Continuity Temporal segmentation is closely related to topical segmentation (Chafe, 1979). Transitions from one topic to another may indicate changes in temporal flow and, therefore,

[2] BoosTexter package (Schapire and Singer, 2000).

identifying such transitions is relevant for temporal segmentation.

We quantify the strength of a topic change by computing a cosine similarity between sentences bordering the proposed segmentation. This measure is commonly used in topic segmentation (Hearst, 1994) under the assumption that change in lexical distribution corresponds to topical change.

Positional Features Some parts of the document are more likely to exhibit temporal change than others. This property is related to patterns in discourse organization of a document as a whole. For instance, a medical case summary first discusses various developments in the medical history of a patient and then focuses on his current conditions. As a result, the first part of the summary contains many short temporal segments. We encode positional features by recording the relative position of a sentence in a document.

Syntactic Features Because our segment boundaries are considered at the clausal level, rather than at the sentence level, the syntax surrounding a hypothesized boundary may be indicative of temporal shifts. This feature takes into account the position of a word with respect to the boundary. For each word within three words of the hypothesized boundary, we record its part-of-speech tag along with its distance from the boundary. For example, NNP_{+1} encodes the presence of a proper noun immediately following the proposed boundary.

5 Learning to Order Segments

Our next goal is to automatically construct a graph that encodes ordering relations between temporal segments. One possible approach is to cast graph construction as a standard binary classification task: predict an ordering for each pair of distinct segments based on their attributes alone. If a pair contains a temporal marker, like *later*, then accurate prediction is feasible. In fact, this method is commonly used in event ordering (Mani et al., 2003; Lapata and Lascarides, 2004; Boguraev and Ando, 2005). However, many segment pairs lack temporal markers and other explicit cues for ordering. Determining their relation out of context can be difficult, even for humans. Moreover, by treating each segment pair in isolation, we cannot guarantee that all the pairwise assignments are consistent with each other and yield a valid TDAG.

Rather than ordering each pair separately, our ordering model relies on global inference. Given the pairwise ordering predictions of a local classifier[3], our model finds a globally optimal assignment. In essence, the algorithm constructs a graph that is maximally consistent with individual ordering preferences of each segment pair and at the same time satisfies graph-level constraints on the TDAG topology.

In Section 5.2, we present three global inference strategies that vary in their computational and linguistic complexity. But first we present our underlying local ordering model.

5.1 Learning Pairwise Ordering

Given a pair of segments (i, j), our goal is to assign it to one of three classes: forward, backward, and null (not connected). We generate the training data by using all pairs of segments (i, j) that belong to the same document, such that i appears before j in the text.

The features we consider for the pairwise ordering task are similar to ones used in previous research on event ordering (Mani et al., 2003; Lapata and Lascarides, 2004; Boguraev and Ando, 2005). Below we briefly summarize these features.

Lexical Features This class of features captures temporal markers and other phrases indicative of order between two segments. Representative examples in this category include domain-independent cues like *years earlier* and domain-specific markers like *during next visit*. To automatically identify these phrases, we provide a classifier with two sets of n-grams extracted from the first and the second segments. The classifier then learns phrases with high predictive power.

Temporal Anchor Comparison Temporal anchors are one of the strongest cues for the ordering of events in text. For instance, medical case summaries use phrases like *two days before admission* and *one day before admission* to express relative order between events. If the two segments contain temporal anchors, we can determine their ordering by comparing the relation between the two anchors. We identified a set of temporal anchors commonly used in the medical domain and devised a small set of regular expressions for their comparison.[4] The corresponding feature has three

[3]The perceptron classifier.

[4]We could not use standard tools for extraction and analysis of temporal anchors as they were developed on the newspaper corpora, and are not suitable for analysis of medical

values that encode preceding, following and incompatible relations.

Segment Adjacency Feature Multiple studies have shown that two subsequent sentences are likely to follow a chronological progression (Bestgen and Vonk, 1995). To encode this information, we include a binary feature that captures the adjacency relation between two segments.

5.2 Global Inference Strategies for Segment Ordering

Given the scores (or probabilities) of all pairwise edges produced by a local classifier, our task is to construct a TDAG. In this section, we describe three inference strategies that aim to find a consistent ordering between *all* segment pairs. These strategies vary significantly in terms of linguistic motivation and computational complexity. Examples of automatically constructed TDAGs derived from different inference strategies are shown in Figure 2.

5.2.1 Greedy Inference in Natural Reading Order (NRO)

The simplest way to construct a consistent TDAG is by adding segments in the order of their appearance in a text. Intuitively speaking, this technique processes segments in the same order as a reader of the text. The motivation underlying this approach is that the reader incrementally builds temporal interpretation of a text; when a new piece of information is introduced, the reader knows how to relate it to already processed text.

This technique starts with an empty graph and incrementally adds nodes in order of their appearance in the text. When a new node is added, we greedily select the edge with the highest score that connects the new node to the existing graph, without violating the consistency of the TDAG. Next, we expand the graph with its transitive closure. We continue greedily adding edges and applying transitive closure until the new node is connected to all other nodes already in the TDAG. The process continues until all the nodes have been added to the graph.

5.2.2 Greedy Best-first Inference (BF)

Our second inference strategy is also greedy. It aims to optimize the score of the graph. The score of the graph is computed by summing the scores of its edges. While this greedy strategy is not guaranteed to find the optimal solution, it finds a reasonable approximation (Cohen et al., 1999).

This method begins by sorting the edges by their score. Starting with an empty graph, we add one edge at a time, without violating the consistency constraints. As in the previous strategy, at each step we expand the graph with its transitive closure. We continue this process until all the edges have been considered.

5.2.3 Exact Inference with Integer Linear Programming (ILP)

We can cast the task of constructing a globally optimal TDAG as an optimization problem. In contrast to the previous approaches, the method is not greedy. It computes the optimal solution within the Integer Linear Programming (ILP) framework.

For a document with N segments, each pair of segments (i, j) can be related in the graph in one of three ways: forward, backward, and null (not connected). Let $s_{i\to j}$, $s_{i\leftarrow j}$, and $s_{i\not\leftrightarrow j}$ be the scores assigned by a local classifier to each of the three relations respectively. Let $I_{i\to j}$, $I_{i\leftarrow j}$, and $I_{i\not\leftrightarrow j}$ be indicator variables that are set to 1 if the corresponding relation is active, or 0 otherwise.

The objective is then to optimize the score of a TDAG by maximizing the sum of the scores of all edges in the graph:

$$\max \sum_{i=1}^{N} \sum_{j=i+1}^{N} s_{i\to j} I_{i\to j} + s_{i\leftarrow j} I_{i\leftarrow j} + s_{i\not\leftrightarrow j} I_{i\not\leftrightarrow j} \quad (1)$$

subject to:

$$I_{i\to j}, I_{i\leftarrow j}, I_{i\not\leftrightarrow j} \in \{0, 1\} \quad \forall\, i, j = 1, \ldots N, i < j \quad (2)$$

$$I_{i\to j} + I_{i\leftarrow j} + I_{i\not\leftrightarrow j} = 1 \quad \forall\, i, j = 1, \ldots N, i < j \quad (3)$$

We augment this basic formulation with two more sets of constraints to enforce validity of the constructed TDAG.

Transitivity Constraints The key requirement on the edge assignment is the transitivity of the resulting graph. Transitivity also guarantees that the graph does not have cycles. We enforce transitivity by introducing the following constraint for every triple (i, j, k):

$$I_{i\to j} + I_{j\to k} - 1 \leq I_{i\to k} \quad (4)$$

If both indicator variables on the left side of the inequality are set to 1, then the indicator variable

text (Wilson et al., 2001).

on the right side must be equal to 1. Otherwise, the indicator variable on the right can take any value.

Connectivity Constraints The connectivity constraint states that each node i is connected to at least one other node and thereby enforces connectivity of the generated TDAG. We introduce these constraints because manually-constructed TDAGs do not have any disconnected nodes. This observation is consistent with the intuition that the reader is capable to order a segment with respect to other segments in the TDAG.

$$(\sum_{j=1}^{i-1} I_{i \not\leftrightarrow j} + \sum_{j=i+1}^{N} I_{j \not\leftrightarrow i}) < N - 1 \qquad (5)$$

The above constraint rules out edge assignments in which node i has null edges to the rest of the nodes.

Solving ILP Solving an integer linear program is NP-hard (Cormen et al., 1992). Fortunately, there exist several strategies for solving ILPs. We employ an efficient Mixed Integer Programming solver *lp_solve*[5] which implements the Branch-and-Bound algorithm. It takes less than five seconds to decode each document on a 2.8 GHz Intel Xeon machine.

6 Evaluation Set-Up

We first describe the corpora used in our experiments and the results of human agreement on the segmentation and the ordering tasks. Then, we introduce the evaluation measures that we use to assess the performance of our model.

6.1 Corpus Characteristics

We applied our method for temporal ordering to a corpus of medical case summaries. The medical domain has been a popular testbed for methods for automatic temporal analyzers (Combi and Shahar, 1997; Zhou et al., 2005). The appeal is partly due to rich temporal structure of these documents and the practical need to parse this structure for meaningful processing of medical data.

We compiled a corpus of medical case summaries from the online edition of The New England Journal of Medicine.[6] The summaries are written by physicians of Massachusetts General Hospital. A typical summary describes an admission status, previous diseases related to the current conditions and their treatments, family history, and the current course of treatment. For privacy protection, names and dates are removed from the summaries before publication.

The average length of a summary is 47 sentences. The summaries are written in the past tense, and a typical summary does not include instances of the past perfect. The summaries do not follow a chronological order. The ordering of information in this domain is guided by stylistic conventions (i.e., symptoms are presented before treatment) and the relevance of information to the current conditions (i.e., previous onset of the same disease is summarized before the description of other diseases).

6.2 Annotating Temporal Segmentation

Our approach for temporal segmentation requires annotated data for supervised training. We first conducted a pilot study to assess the human agreement on the task. We employed two annotators to manually segment a portion of our corpus. The annotators were provided with two-page instructions that defined the notion of a temporal segment and included examples of segmented texts. Each annotator segmented eight summaries which on average contained 49 sentences. Because annotators were instructed to consider segmentation boundaries at the level of a clause, there were 877 potential boundaries. The first annotator created 168 boundaries, while the second — 224 boundaries. We computed a Kappa coefficient of 0.71 indicating a high inter-annotator agreement and thereby confirming our hypothesis about the reliability of temporal segmentation.

Once we established high inter-annotator agreement on the pilot study, one annotator segmented the remaining 52 documents in the corpus.[7] Among 3,297 potential boundaries, 1,178 (35.7%) were identified by the annotator as segment boundaries. The average segment length is three sentences, and a typical document contains around 20 segments.

6.3 Annotating Temporal Ordering

To assess the inter-annotator agreement, we asked two human annotators to construct TDAGs from

[5] http://groups.yahoo.com/group/lp_solve

[6] http://content.nejm.org

[7] It took approximately 20 minutes to segment a case summary.

five manually segmented summaries. These summaries consist of 97 segments, and their transitive closure contain a total of 1,331 edges. We computed the agreement between human judges by comparing the transitive closure of the TDAGs. The annotators achieved a surprisingly high agreement with a Kappa value of 0.98.

After verifying human agreement on this task, one of the annotators constructed TDAGs for another 25 summaries.[8] The transitive reduction of a graph contains on average 20.9 nodes and 20.5 edges. The corpus consists of 72% forward, 12% backward and 16% null segment edges inclusive of edges induced by transitive closure. At the clause level, the distribution is even more skewed — forward edges account for 74% edges, equal for 18%, backward for 3% and null for 5%.

6.4 Evaluation Measures

We evaluate temporal segmentation by considering the ratio of correctly predicted boundaries. We quantify the performance using F-measure, a commonly used binary classification metric. We opt not to use the P_k measure, a standard topical segmentation measure, because the temporal segments are short and we are only interested in the identification of the exact boundaries.

Our second evaluation task is concerned with ordering manually annotated segments. In these experiments, we compare an automatically generated TDAG against the annotated reference graph. In essence, we compare edge assignment in the transitive closure of two TDAGs, where each edge can be classified into one of the three types: forward, backward, or null.

Our final evaluation is performed at the clausal level. In this case, each edge can be classified into one of the four classes: forward, backward, equal, or null. Note that the clause-level analysis allows us to compare TDAGs based on the automatically derived segmentation.

7 Results

We evaluate temporal segmentation using leave-one-out cross-validation on our corpus of 60 summaries. The segmentation algorithm achieves a performance of 83% F-measure, with a recall of 78% and a precision of 89%.

[8] It took approximately one hour to build a TDAG for each segmented document.

To evaluate segment ordering, we employ leave-one-out cross-validation on 30 annotated TDAGs that overall contain 13,088 edges in their transitive closure. In addition to the three global inference algorithms, we include a majority baseline that classifies all edges as forward, yielding a chronological order.

Our results for ordering the manually annotated temporal segments are shown in Table 1. All inference methods outperform the baseline, and their performance is consistent with the complexity of the inference mechanism. As expected, the ILP strategy, which supports exact global inference, achieves the best performance — 84.3%.

An additional point of comparison is the accuracy of the pairwise classification, prior to the application of global inference. The accuracy of the local ordering is 81.6%, which is lower than that of ILP. The superior performance of ILP demonstrates that accurate global inference can further refine local predictions. Surprisingly, the local classifier yields a higher accuracy than the two other inference strategies. Note, however, the local ordering procedure is not guaranteed to produce a consistent TDAG, and thus the local classifier cannot be used on its own to produce a valid TDAG.

Table 2 shows the ordering results at the clausal level. The four-way classification is computed using both manually and automatically generated segments. Pairs of clauses that belong to the same segment stand in the equal relation, otherwise they have the same ordering relation as the segments to which they belong.

On the clausal level, the difference between the performance of ILP and BF is blurred. When evaluated on manually-constructed segments, ILP outperforms BF by less than 1%. This unexpected result can be explained by the skewed distribution of edge types — the two hardest edge types to classify (see Table 3), backward and null, account only for 7.4% of all edges at the clause level.

When evaluated on automatically segmented text, ILP performs slightly worse than BF. We hypothesize that this result can be explained by the difference between training and testing conditions for the pairwise classifier: the classifier is trained on manually-computed segments and is tested on automatically-computed ones, which negatively affects the accuracy on the test set. While all the strategies are negatively influenced by this discrepancy, ILP is particularly vulnerable as it relies

Algorithm	Accuracy
Integer Linear Programming (ILP)	**84.3**
Best First (BF)	78.3
Natural Reading Order (NRO)	74.3
Baseline	72.2

Table 1: Accuracy for 3-way ordering classification over manually-constructed segments.

Algorithm	Manual Seg.	Automatic Seg.
ILP	**91.9**	84.8
BF	91.0	**85.0**
NRO	87.8	81.0
Baseline	73.6	73.6

Table 2: Results for 4-way ordering classification over clauses, computed over manually and automatically generated segments.

on the score values for inference. In contrast, BF only considers the rank between the scores, which may be less affected by noise.

We advocate a two-stage approach for temporal analysis: we first identify segments and then order them. A simpler alternative is to directly perform a four-way classification at the clausal level using the union of features employed in our two-stage process. The accuracy of this approach, however, is low — it achieves only 74%, most likely due to the sparsity of clause-level representation for four-way classification. This result demonstrates the benefits of a coarse representation and a two-stage approach for temporal analysis.

8 Conclusions

This paper introduces a new method for temporal ordering. The unit of our analysis is a temporal segment, a fragment of text that maintains temporal coherence. After investigating several inference strategies, we concluded that integer linear programming and best first greedy approach are valuable alternatives for TDAG construction.

In the future, we will explore a richer set of constraints on the topology on the ordering graph. We will build on the existing formal framework (Fikes et al., 2003) for the verification of ordering consistency. We are also interested in expanding our framework for global inference to other temporal annotation schemes. Given a richer set of temporal relations, the benefits from global inference can be even more significant.

Algorithm	Forward	Backward	Null
ILP	**92.5**	**45.6**	**76.0**
BF	91.4	42.2	74.7
NRO	87.7	43.6	66.4

Table 3: Per class accuracy for clause classification over manually computed segments.

Acknowledgments

The authors acknowledge the support of the National Science Foundation and National Institute of Health (CAREER grant IIS-0448168, grant IIS-0415865). Thanks to Terry Koo, Igor Malioutov, Zvika Marx, Benjamin Snyder, Peter Szolovits, Luke Zettlemoyer and the anonymous reviewers for their helpful comments and suggestions. Any opinions, findings, conclusions or recommendations expressed above are those of the authors and do not necessarily reflect the views of the NSF or NIH.

References

James F. Allen. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

Regina Barzilay, Noemie Elhadad, and Kathleen McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Yves Bestgen and Wietske Vonk. 1995. The role of temporal segmentation markers in discourse processing. *Discourse Processes*, 19:385–406.

Branimir Boguraev and Rie Kubota Ando. 2005. Timeml-compliant text analysis for temporal reasoning. In *Proceedings of IJCAI*, pages 997–1003.

Wallace Chafe. 1979. The flow of thought and the flow of language. In Talmy Givon, editor, *Syntax and Semantics: Discourse and Syntax*, volume 12, pages 159–182. Academic Press.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the NAACL*, pages 132–139.

William Cohen, Robert Schapire, and Yoram Singer. 1999. Learning to order things. *Journal of Artificial Intelligence*, 10:243–270.

Carlo Combi and Yuval Shahar. 1997. Temporal reasoning and temporal data maintenance in medicine: Issues and challenges. *Computers in Biology and Medicine*, 27(5):353–368.

Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. 1992. *Intoduction to Algorithms*. The MIT Press.

David R. Dowty. 1986. The effects of aspectual class on the temporal structure of discourse: Semantics or Pragmatics? *Linguistics and Philosophy*, 9:37–61.

R. Fikes, J. Jenkins, and G. Frank. 2003. A system architecture and component library for hybrid reasoning. Technical report, Stanford University.

Marti Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the ACL*, pages 9–16.

Chung Hee Hwang and Lenhart K. Schubert. 1992. Tense trees as the "fine structure" of discourse. In *Proceedings of the ACL*, pages 232–240.

Mirella Lapata and Alex Lascarides. 2004. Inferring sentence-internal temporal relations. In *Proceedings of HLT-NAACL*, pages 153–160.

Alex Lascarides and Nicholas Asher. 1993. Temporal interpretation, discourse relations, and commonsense entailment. *Linguistics and Philosophy*, 16:437–493.

Alex Lascarides and John Oberlander. 1993. Temporal connectives in a discourse context. In *Proceeding of the EACL*, pages 260–268.

Inderjeet Mani, Barry Schiffman, and Jianping Zhang. 2003. Inferring temporal ordering of events in news. In *Proceeding of HLT-NAACL*, pages 55–57.

Mark Moens and Mark J. Steedman. 1987. Temporal ontology in natural language. In *Proceedings of the ACL*, pages 1–7.

Rebecca J. Passonneau. 1988. A computational model of the semantics of tense and aspect. *Computational Linguistics*, 14(2):44–60.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, David Day, Lissa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth Sundheim. 2003. The timebank corpus. *Corpus Linguistics*, pages 647–656.

Hans Reichenbach. 1947. *Elements of Symbolic Logic*. Macmillan, New York, NY.

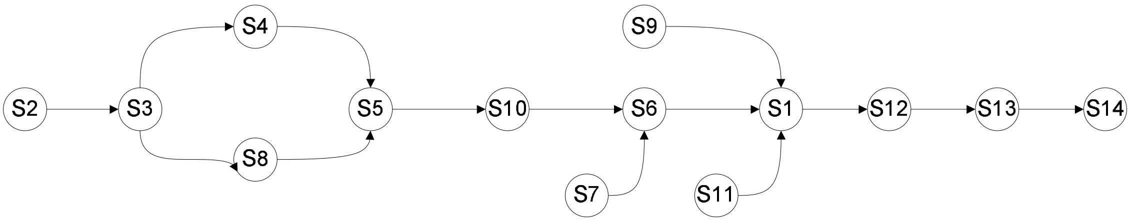
Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Bonnie L. Webber. 1987. The interpretation of tense in discourse. In *Proceedings of the ACL*, pages 147–154.

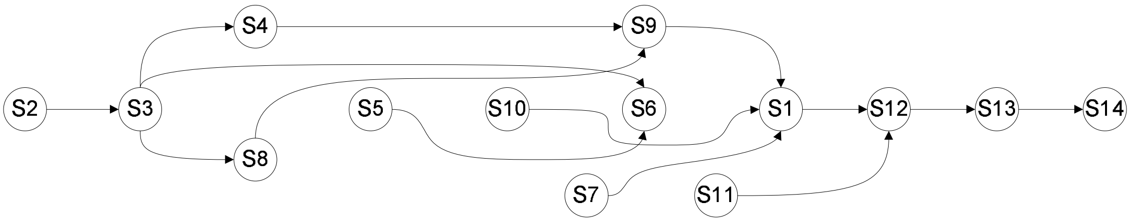
Bonnie L. Webber. 1988. Tense as discourse anaphor. *Computational Linguistics*, 14(2):61–73.

George Wilson, Inderjeet Mani, Beth Sundheim, and Lisa Ferro. 2001. A multilingual approach to annotating and extracting temporal information. In *Proceedings of the ACL 2001 Workshop on Temporal and Spatial Information Processing*, pages 81–87.

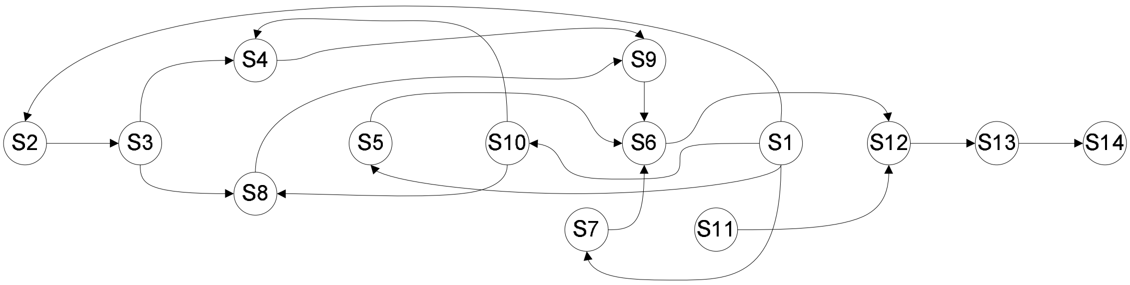
Li Zhou, Carol Friedman, Simon Parsons, and George Hripcsak. 2005. System architecture for temporal information extraction, representation and reasoning in clinical narrative reports. In *Proceedings of AMIA*, pages 869–873.



(a) Reference TDAG



(b) ILP generated TDAG with an accuracy of 84.6%



(c) BF generated TDAG with an accuracy of 71.4%; NRO produces the same graph for this example.

S1	A 32-year-old woman was admitted to the hospital because of left subcostal pain. . .
S2	The patient had been well until four years earlier,
S3	when she began to have progressive, constant left subcostal pain, with an intermittent increase in the temperature to 39.4°C, anorexia, and nausea. The episodes occurred approximately every six months and lasted for a week or two;
S4	they had recently begun to occur every four months.
S5	Three months before admission an evaluation elsewhere included an ultrasonographic examination, a computed tomographic (CT) scan of the abdomen. . .
S6	Because of worsening pain she came to this hospital.
S7	The patient was an unemployed child-care worker. She had a history of eczema and of asthma. . .
S8	She had lost 18 kg in weight during the preceding 18 months.
S9	Her only medications were an albuterol inhaler, which was used as needed,
S10	and an oral contraceptive, which she had taken during the month before admission.
S11	There was no history of jaundice, dark urine, light stools, intravenous drug abuse, hypertension, diabetes mellitus, tuberculosis, risk factors for infection with the human immunodeficiency virus, or a change in bowel habits. She did not smoke and drank little alcohol.
S12	The temperature was 36.5°C, the pulse was 68, and the respirations were 16. . .
S13	On examination the patient was slim and appeared well. . . An abdominal examination revealed a soft systolic bruit. . . and a neurologic examination was normal. . .
S14	A diagnostic procedure was performed.

(d) An example of a case summary

Figure 2: Examples of automatically constructed TDAGs with the reference TDAG and text.