Natural Language Engineering **3** (2/3): 191–213. Printed in the United Kingdom

© 1997 Cambridge University Press

Recovering nonlinearly distributed knowledge: Computing discourse structure in factual reports

ŁUCJA IWAŃSKA

Department of Computer Science, Wayne State University, Detroit, MI 48202, USA
e-mail: lucja@cs.wayne.edu

(*Received 11 June 1997*)

Abstract

This paper presents a new type of nonlinear discourse structure found to be very common in free English texts. This structure reflects nonlinear presentation of the information and knowledge conveyed by the texts. It is argued that such nonlinearity is representationally and informationally advantageous because it allows one to create smaller, more compact texts. The paper presents a heuristics-based, relatively domain-independent algorithm for computing this new text structure. The paper discusses good quantitative and qualitative performance of the algorithm, and presents the results of the extensive tests on a large volume of free English texts.

1 Introduction

1.1 Discourse structure – nonlinear knowledge distribution

To communicate knowledge about entities of interest (objects, individuals, events and their sets) and fully describe complex relations among them, one usually needs far more than a single sentence to do so. In the case of newspaper reporting, authors typically write one–two page, multi-sentence, multi-paragraph articles.

Reported entities frequently exhibit a number of similarities, including the same general location, similar time frames and the same persons involved. For example, an article may report a series of fires in a particular geographical region, such as Northern California. Reported entities are often characterized as a group. For example, a total property damage after a natural disaster in a number of different locations characterizes collectively the areas affected.

To avoid redundancy in providing such common information in a linear, entity-by-entity fashion, or to save some space, authors frequently opt for a non-linear distribution of information, with some units of text stating the known commonalities and similarities, and other units of text providing additional (more specific and other) information in a linear, entity-by-entity fashion.

The resulting nonlinear structure of text, with some segments containing information applicable to all other segments, allows one to shorten the report, or equivalently,

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

to maximize information content of a fixed-size report. Additionally, such a nonlinear structure increases text coherence. Commonalities are stated explicitly and serve as links among multi-sentence, multi-paragraph units of texts.

1.2 Discourse processing – recovering nonlinearly distributed knowledge

Recovering such nonlinearly presented knowledge is very important. It allows one to properly distinguish individual entities, to fully understand stated relations among them, and to correctly understand their common and individual characteristics. Such a computation, by a person or a program, is one critical aspect of discourse processing, the capability of understanding connected text that comprises phenomena usually not arising in processing a single sentence. One example of a relatively well understood text-level phenomenon is 'definite anaphora'.

This paper presents a discourse processing algorithm that allows one to recover two types of nonlinear structures frequently found in free English texts, factual reports such as newspaper articles in the *Wall Street Journal* and *Time Magazine*. Computing such structures is crucial for constructing a knowledge base of entities described in the text, and for properly distributing information about these entities provided in different segments of text.

1.3 Approach: Engineering, not so theoretical solution

This paper reports two main contributions.

1. **New type of discourse structure**
   I have identified a new type of discourse structure common in large corpora of free English texts. The attractiveness of this structure stems from its representational and informational advantage – structuring texts in this particular fashion allows one to create smaller, more compact texts.
   The motivation for this new discourse structure is similar to the domain-independent, rhetorical relation-based discourse structure (Mann and Thompson 1991) and quite different from the intention-based structure of Grosz and Sidner (1986). It is interesting because Moser and Moore (1996) argue that these two existing theories of discourse structure are basically equivalent.
2. **Algorithm for computing this discourse structure**
   I have designed, implemented and tested an algorithm for computing this discourse structure in a large corpus of English texts in two domains, *terrorist attacks* and *joint venture*. It is also demonstrated that this algorithm performs well on some complicated dialogs.
   The algorithm is fully implemented as a discourse processing module written in Common Lisp, and was extensively tested. This module can be used in a stand-alone mode; for example, for investigating the discourse structure of various texts. It can also be used as a module of a natural language processing system performing some task that critically depends upon recovering nonlinearly structured information in the text; for example, for producing text-level representation of the knowledge about entities provided by the text.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

1.3.1 Goal: Reducing domain-dependency, improving performance

The goal was to develop a relatively domain-independent discourse processing module that would augment a system implementing a domain-and-expectation-driven, slot-filler approach to text understanding (Iwańska 1991; Iwańska *et al.* 1991). The only way that this particular system could make sense out of multiple-sentence texts was through the domain-specific concepts that had to be known to the system *a priori*. The system lacked any other means of connecting individual sentences – even a rudimentary anaphora resolution was not available.

One problem with this domain-driven approach was that it did not correctly distinguish individual events in articles reporting multiple events. As a result, it would produce too few templates and incorrectly place information about one entity in the template of another. This was particularly true when information about reported entities was partial, and when a lot was known about some entities and very little about the others. This domain-driven approach almost always missed nonlinearly distributed information. At best, it would apply information from a segment with commonalities to the segment immediately following it. As a result, many slots in the templates produced were unnecessarily missing their fillers.

1.3.2 Algorithm performs well on large corpora

While perceived originally as an engineering rather than a theoretical solution, my approach to computing certain common types of discourse structure offers a number of attractive characteristics. It has been demonstrated to work well on large volumes – several hundreds – of free English texts. In contrast, those few algorithms that are reported in the non-engineering literature are based on a very small number – a few tens – of examples, and are not implemented (e.g. Morris and Hirst 1991).

Research on discourse structure is still in its infancy. There is little agreement about the notion of text segment boundary. Psychological data are rather scarce. For example, Litman and Passonneau (1993) discuss their experimental findings about text boundary based on a single text and four human subjects.

1.3.3 Theoretical vs. engineering: Good vs. bad?

The biggest difference between a theoretical solution and an engineering solution is their respective (in)capability of capturing the entire phenomenon via a simple, elegant insight leading to a simple, effective code. An engineering solution is one that works well, say above 70% in terms of some relevant quantitative measure, one that captures some good portion of the reality and possibly misses some important aspects of the phenomenon, one that does not readily offer a higher-level explanation, and one which is typically not very pretty or simple. A theoretical solution is one that solves or seriously attempts to solve the entire problem, and if implemented, results in a near-perfect, high 90%s performance, is based on some great insight into a phenomenon, which results in its typical simplicity and elegance.

In the natural language processing and computational linguistics literature, there is a definite bias toward preferring 'theoretical' solutions. This is particularly true

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

for text-level phenomena such as discourse structure. Highly explanatory work that aimes at capturing the entire phenomenon, most often unimplemented, is 'good'. Engineering solutions are 'bad'.

Rationally speaking, this bias is strange because untested, unimplemented theories that might be (and very often are) relying on some unreasonable assumptions are equally bad as little-explanation engineering solutions.

1.3.4 Engineering and theoretical aspects of the approach

Some of the engineering aspects of this new approach are apparent:

1. The new algorithm is neither very simple nor particularly elegant; in fact, some parts of the code that implement it are downright ugly – with many if-conditions designed to capture specific, sometimes *ad hoc*, cases.
2. Some things work well, but I cannot explain why exactly they work in a general, meaningful way that would reveal some deep characteristics of natural language. Producing some of the explanations discussed in this paper took me much longer than the actual work of designing, implementing and testing the algorithm.
3. Preserving a good performance with even seemingly small changes is not always easy.

There are two main theoretical insights of this new solution:

1. For the identified discourse structures, a justification is offered based on information gain and a representational utility. These discourse structures allow one to maximize information-content of fixed-size texts. This justification is rather general and should hold in any domain.
2. The new algorithm embodies a number of highly reliable, easily computable syntactic components for identifying (computing) such discourse structures, which gives hope for truly domain-independent processing.

In summary, this mostly engineering solution embodies some healthy theoretical elements. The combination of its performance on a large volume of texts, explanations and justifications of the ideas behind it, as well as good alignment of the human- and algorithm-produced text segmentations, collectively demonstrate that the algorithm captures some good portion of the largely unknown discourse structure reality. Such a reasonable approximation is often a (very good) beginning of a good theory.

1.4 More related work

Existing theories of discourse structure and some text-level phenomena such as focus appeared (and still do) potentially relevant for computing discourse structures identified here, but turned out to be (and still appear to be) basically useless, because, first, they make a totally unrealistic (for a large corpus) assumption that sentence-level parses and interpretations are available, and second, they were typically developed on an artificial, tiny corpus, and were not verified experimentally.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

1.4.1 Closer to intuitive, somewhat precise and verifiable definition of text segment boundary

By providing an intuitive, somewhat precise and, most of all, verifiable definition of the text segment boundary, some more light is shed on the text boundary mystery. This is important because there is very little agreement in the literature as to what a text boundary is and what it means (Grosz and Sidner 1986; Morris and Hirst 1991; Litman and Passonneau 1993).

My approach is based on the observation of the phenomenon that can be justified and explained in terms of specific representational, information-based advantages. Texts exhibit certain nonlinear structure, because this structure allows one to convey more information in a fixed-size text and make this information more understandable.

This explanation of the representational utility of text structure should be seriously considered by computational models of natural language aiming to truthfully represent the meaning of text. Such text-level structured representation of natural language appears to be advantages over flat, linear representation.

1.4.2 Appears to do what people do

There is an additional piece of independent experimental evidence in support for my approach to computing discourse structure. In their study of segmentation in spontaneous speech, Litman and Passonneau (1993) asked human subjects to identify boundaries in the pear story (Chafe 1980), a celebrated example of a complicated discourse structure (discussed later in the paper). The boundaries produced by the human subjects coincide almost exactly with the boundaries produced by my new approach (personal communication).

1.4.3 Why (not) recursive structure?

Texts in the corpora considered did not exhibit recursive structures. In contrast, a very short, much shorter than a typical newspaper article, dialog discussed in Grosz and Sidner (1986) is argued to have a highly recursive structure. There are a number of reasons that may explain this lack of recursion in factual reports.

First, the task of factual reporting is to convey new, most probably unknown, information to the readers. Writers and readers do not share this knowledge; if they did, there would be no story. It can then be reasonably assumed that (much) less common knowledge exists between readers and writers than between two professionals engaging in a dialog while collaborating on a task in their areas of expertise.

Second, reports may have simple, nonrecursive structures because readers do not have an opportunity to ask the author clarifying questions. This is a very different situation from a dialog in which the participants are able to clarify their misunderstandings by requesting additional information.

Little common knowledge and the possibility of various misunderstandings may both be responsible for incorrect recovery of the underlying, intended structure of natural language utterances. This in turn would result in such problems as incorrect

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

reference resolution. It is plausible to conjecture that when the two dangers are not likely, i.e. when lots of common knowledge is shared and misunderstandings are not likely, then utterances may be deeply recursively structured. But in cases when one of these dangers is serious, for example, when communicated knowledge is likely not to be shared, then the utterance structure is simple and nonrecursive, which minimizes the possibility of structure recovery errors.

1.4.4 Boundary for anaphora or anaphora for boundary?

One of the heuristics incorporated in the new algorithm approximates resolution of definite anaphora. It is resolving anaphoric noun phrases that allows the algorithm to identify text segment boundaries. This is the exact opposite from Grosz and Sidner (1986), who argue that it is knowing the discourse boundaries that facilitates (simplifies) computation of anaphora resolution.

1.4.5 Whose structure: Writer's or reader's?

The capability of identifying entities of interest as specified by the key phrases allows the users to concentrate on those fragments of text that contain information useful for them. Regardless of the writer-intended structure of the input text, users can impose their own structure upon the text and create their knowledge bases accordingly.

This capability of imposing structure reflecting user interests is a unique capability of the new algorithm. It is important because it accounts for the fact that one is constantly faced with the necessity of extracting useful information from the writings of people with a different perspective or interests. When the readers' interests coincide with the writer's interests, the computed text structure often coincides with the text structure intended by the writer. This happens when a story is about the entities of interest, in which case the activators have a lot in common with the notion of the topic of a story. Both sample stories presented later are of this kind.

This paper is organized as follows. Section 2 discusses nonlinear discourse structures identified, provides two examples of newspaper articles exhibiting such structures and gives details about the test and development data. Section 3 presents the algorithm for computing these discourse structures, and section 4 discusses the algorithm and system performance on a sample example from the test and development corpus. Section 5 discusses strengths and weaknesses of the approach and sketches some possible extensions. Section 6 contains conclusions.

2 Nonlinear text structure: Phenomenon

2.1 Test and development data

The test and development data comprised two corpora:

Corpus 1: *terrorist attacks* domain

300 English texts from the MUC-3 corpus (Sundheim 1991) with the following characteristics: it contains 1700 news wire reports of 16 different types,

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

including articles, interviews, editorial reports and communiques; its estimated vocabulary is 18,000 words, the average sentence length 27 words, and the average article length 12 sentences.

Corpus 2: *joint venture* domain

100 English texts in the MUC-4 corpus (MUC 1992).

Sample texts 1 and 2 show sample texts from these corpora.

<DOCNO> 0063 </DOCNO>
<DD> JANUARY 8, 1991, TUESDAY </DD>
<SO> Copyright (c) 1991 Kyodo News Service </SO>
<TXT>

NISSEI BUILD KOGYO CO. SAID TUESDAY IT WILL LAUNCH JOINT VENTURES IN ITALY, CANADA, SOUTH KOREA, AND TAIWAN TO CONSTRUCT HIGH-RISE PARKING FACILITIES.

THE KANAZAWA-BASED COMPANY SAID THE VENTURES WILL BE EQUALLY OWNED, WITH NISSEI PROVIDING ELEVATORS, CONTROL SYSTEMS, AND TURNTABLES. LOCAL PARTNERS WILL HANDLE PRODUCTION OF STEEL FRAMES, ASSEMBLY, AND MAINTENANCE.

THE COMPANY SAID ITS JOINT VENTURE IN CANADA PLANS TO EXPORT FACILITIES TO SAN FRANCISCO AND OTHER CITIES ON THE U.S. WEST COAST.

THE COMPANY SAID THE NEW COMPANIES TO BE ESTABLISHED WILL BE IMER NISSEI AND IMER INTERNATIONAL IN POGGIBONSI, ITALY.

THE OTHERS WILL BE THE BALSHINE FOUNDATION AND GEORGE THE THIRD AND SON IN VANCOUVER, CANADA= SAMICK NISSEI CO. AND WOOSUNG HEAVY INDUSTRIAL CO. IN INCHON, SOUTH KOREA= AND TAIWAN NISSEI, WHICH WILL BE FORMED BY NISSEI AND A GROUP OF MEDIUM AND SMALL ENTERPRISES IN TAIWAN.
</TXT>
</doc>

Sample Text 1. 7-sentence, 6-paragraph newspaper story in the *joint venture* domain.

TST2-MUC3-0004

LA PAZ, 20 DEC 89 (EFE) – [TEXT] A POWERFUL DYNAMITE CHARGE TODAY EXPLODED NEAR THE AMERICAN EMBASSY IN LA PAZ. THE POLICE REPORTED THAT THE EXPLOSION CAUSED CONSIDERABLE DAMAGE TO ALL BUILDINGS ON THE BLOCK.

THE EXPLOSION OCCURRED AT 2130 (0130 GMT) ON THURSDAY [AS RECEIVED]. IT DAMAGED WINDOWS OF THE BUILDINGS ON THE BLOCK.

JOSE PARADA GRANDY, THE BOLIVIAN POLICE CHIEF, TOLD EFE THAT AN UNIDENTIFIED PERSON STEPPED OUT OF A VEHICLE AND PLACED A PACKAGE IN ONE OF THE PLANT POTS ON JUAN DE LA RIVA STREET, A FEW METERS FROM THE U.S. EMBASSY IN DOWNTOWN LA PAZ.

PARADA SAID: "THE VEHICLE LEFT AND IMMEDIATELY THE CHARGE WENT OFF CAUSING CONSIDERABLE DAMAGE."

THE "ZARATE WILLKA ARMED FORCES OF LIBERATION," A GROUP WITH AN INDIAN BAKCGROUND, HAS CLAIMED RESPONSIBILITY FOR THE ATTACK IN A COMMUNIQUE THAT WAS SENT TO SOME MEDIA.

THE NOTE, SIGNED WITH THE INITIALS OF THE "ZARATE WILLKA" GROUP, STATES: "IN DEFENSE OF THE SOVEREIGNTY OF THE PANAMANIAN PEOPLE. DEATH TO THE INVADER. OUR HATRED IS RELENTLESS; OUR WAR IS TO DEATH."

THE "ZARATE WILLKA" GROUP CLAIMED RESPONSIBILITY IN AUGUST 1988 FOR THE ATTACK AGAINST THEN U.S. SECRETARY OF STATE GEORGE SHULTZ AND THE KILLING OF TWO MORMON MISSIONARIES.

TODAY'S BOMB EXPLOSION DAMAGED THE SHELVES OF A BOOK STORE, THE "PEOPLE'S PERUVIAN BANK," THE STATE BANK [NOT FURTHER IDENTIFIED]; THE MARISCAL BALLIVIAN BUILDING, AND OTHER SHOPS, ALL OF WHICH ARE LOCATED NEAR THE U.S. EMBASSY.

AFTER THE EXPLOSION, A GROUP OF U.S. AGENTS CALLED OFF AN INVESTIGATION AT THE SCENE OF THE INCIDENT BECAUSE SOME PEOPLE SHOUTED AT THEM, "YANKEES, OUT OF PANAMA."

THE U.S. EMBASSY PRESS ATTACHE, BRUCE WARTON, SAID TODAY THAT "IT IS OBVIOUSLY A DESPICABLE ACTION THAT DOES MORE HARM TO BOLIVIANS THAN TO THE U.S. EMBASSY, BECAUSE WE HAVE SUFFERED NO DAMAGES."

Sample Text 2. 13-sentence, 10-paragraph newspaper story in the *terrorist attacks* domain.

2.2 Nonlinear text structure

Two types of nonlinear discourse structure common in factual reporting can be identified:

1. *SI-structure*, or *Shared Information* structure.
2. *E-structure*, or *Embedded* structure.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

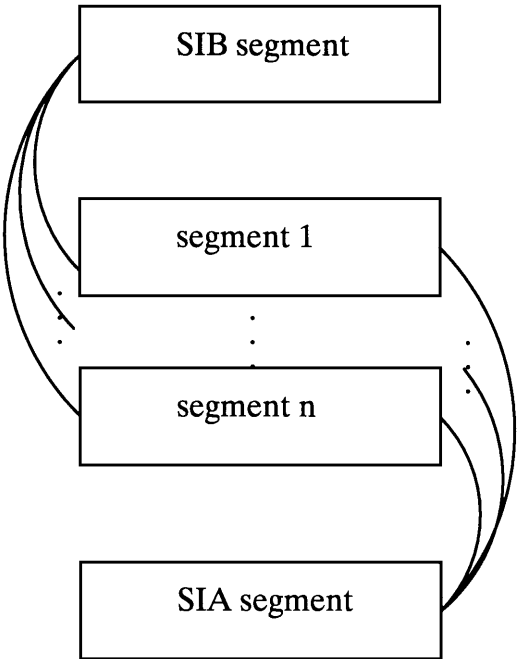


Fig. 1. A SI-structure of text reflects a nonlinear distribution of communicated knowledge: the SIB segment contains information common to the set of entities $1 \ldots n$ described in the subsequent n segments; the SIA segment contains information pertinent to the set of entities described in the earlier n segments. Links connect segments whose content should be combined.

2.2.1 SI-structure

The first type of nonlinear structure identified is the SI-structure, shown in Figure 1. It consists of the following multiple-sentence, multiple-paragraph ordered units of text:

1. *SIB-segment*, or Shared Information Before Segment – a segment of text containing information shared by a set of entities described later in the text;
2. A number of segments following the SIB segment that describe individual entities;
3. *SIA-segment*, or Shared Information After Segment – a segment of text following the individual segments and containing information pertinent to the entities described earlier in the text.

The SIB segment contains information common to the entities described later in the text. This information may be a distributive property that is true of each individual entity. An example of such a distributive property is the time of a series of attacks. The information contained in the SIB segment may also be a cumulative property that is true of the set of entities, but not necessarily of each individual entity. An example of such a cumulative property is the total damage caused by a number of attacks.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

The SIB segment sometimes coincides with the introduction of the story (van Dijk and Kintsch 1983). The SIA segment is usually some kind of summary and often contains a plural definite anaphora.

Information contained in the SIB and SIA segments often acts as a shorthand that allows one to state the commonalities of some entities once, instead of repeating them a number of times. This shorthand function of the SIB and SIA segments is somewhat similar to the function of the sentence-level universal quantification. To answer certain questions about the entities described, one needs to correctly combine information provided by the SIB and SIA segments with the information in all the individual segments.

The type of this combination depends upon the type of commonalities expressed by the SIB and SIA segments. Distributive properties such as general location and time frame call for conjunction-like combination. Cumulative properties such as various totals should not be distributed over individual segments. Commonalities that act as contextual variables (e.g. a general description of local offices) must be evaluated in the context of an individual forthcoming segment. Such evaluations often require nontrivial computation.

All these functions are illustrated by the SIB segment for the *joint venture* Sample Text 1. Its SIB segment spans the first two paragraphs and informs readers that all the joint ventures described later in the story are launched by a single company (distributive property), that these joint ventures will be equally owned (cumulative property) and that local partners will have similar responsibilities (contextual variable).

A full SI structure with both the SIB and SIA segments is quite rare. Its special forms in which only one segment with shared information, either the SIB or SIA segment, is present, are very common. The stories with no shared information segment are also numerous. And finally, there are many stories that report a single entity, i.e. do not exhibit any such structure. This mix of stories with no, little and a lot of structure greatly complicates the problem of computing the actual structure for a given text.

2.2.2 E-structure

The second type of nonlinear discourse structure identified is the E-structure shown in Figure 2. It consists of a segment of text describing an entity with an embedded segment of text describing another entity. The E-structure is common because new entities are often compared with the old, known ones. Digressions frequently result in E-structures. In general, one observes such embeddings when a relevant property is shared by the new and some known entities.

In the *terrorist attacks* Sample Text 1, the old attacks are mentioned while describing the new attack because the same organization claimed the responsibility for these attacks.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

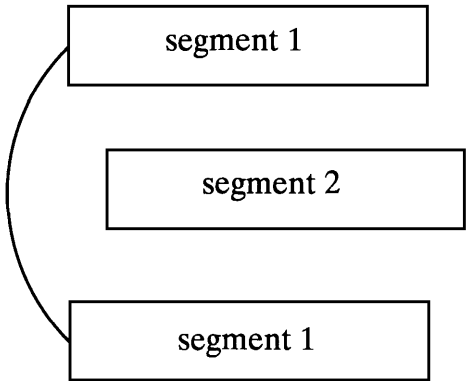


Fig. 2. An E-structure of text reflects a nonlinear distribution of communicated knowledge: a text segment that corresponds to one entity is embedded in the segment describing another entity. Information in linked segments needs to be combined appropriately.

2.3 Methodology

The methodology was somewhat rigorous, but not very rigorous, mainly due to time constraints and the fact that industrial research setting had a strong bias toward engineering, and not theoretically impeccable solutions. The 300-texts **Corpus 1** contained roughly 200 articles relevant for the *terrorist attacks* domain, and among them, about 120 texts contained the SI- and E-structures, and the other 80 were a single-entity, no structure texts.

The presence and particular type (segments plus links) structures were first identified on-paper by one person, and then verified by two different people. There was an almost perfect agreement with respect to whether a text does or does not contain a SI- or E-structure, with only one problematic and excluded text.

The algorithm was then tested and its quantitative performance in both domains estimated to be in the 85–95% range.

For the *terrorist attacks* domain:

False positives, i.e. single-entity, domain-relevant texts for which the module incorrectly outputted a multi-segment structure; There were very few (about five) false positives, a mixture of domain-irrelevant and no-structure texts. Even very long for this corpus, three pages or so, texts were correctly understood as describing a single entity.

False negatives, i.e. multi-entity texts for which the module failed to identify different text segments and their links; There were very few (about three) false negatives.

Fully correct structures, i.e. structures whose segments and links agreed with the human-marked ones. Most (about 110) computed structures were in this category;

Partially correct structures, i.e. texts for which the program correctly identified some, but not all, text segments and recovered some, but not all, links. There were about 20 texts for which most structure elements were correctly computed; occasionally, the program would produce one or two too small (unnecessarily split) segments, or one or two too large (not split enough) segments.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

The performance for the subsequently slightly extended algorithm in the *joint venture* domain was much better.

3 Algorithm

3.1 High-level description

The algorithm presented in this paper computes two types of nonlinear discourse structure identified. It was originally designed for and tested on the **Corpus 1**, *terrorist attacks* domain, and then generalized to and tested on the **Corpus 2**, *joint venture* domain.

This algorithm does not crucially depend upon the correct sentence-level parsing or interpretation of the input sentences. This is important because in most free texts, sentential-level parses are basically unobtainable even by the best state-of-the-art parsers.

The algorithm first identifies basic segments that describe individual entities (single entities or their groups), and then computes links connecting these segments. As a side-effect, irrelevant portions of the text can be filtered out.

A segment is defined as an ordered set of those text fragments that describe the same entity. A text fragment consists of some number of consecutive sentences; it may span more than one paragraph. Additional text fragmentation may result from filtering out irrelevant text. Unless there is filtering of irrelevant portions of text, for the SI-structure, a segment consists of a single text fragment, and for the E-structure, the enclosing segment consists of two fragments of text.

The boundary between two consecutive segments is a related concept (figures 3 and 4). It falls between the last sentence of the first segment and the first sentence of the second segment. Subsequently, the algorithm merges certain basic segments into a single segment. The strategy for continuation and merging for such segments is much more conservative.

3.2 Underlying concepts – some definitions

The algorithm utilizes syntactic, semantic and pragmatic information, a mixture of domain-specific and domain-independent input and processing. It relies on the following concepts:

Primary activator: a key phrase, a typically simple expression that identifies an entity of interest as defined by the task in a given domain; For example, the expression *set the building on fire* is a primary activator that identifies an *arson* event important in the *terrorist attacks* domain. The algorithm uses primary activators for deciding whether a new segment (the description of a new entity) starts or the previous segment (the description of the previous entity) continues.

Secondary activator: a key phrase, a regular expression indicative of the presence of the domain-relevant information. For example, the expression *some damage* is a secondary activator for an attack event in the *terrorist attacks* domain.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

Secondary activators are used to decide the actual placement of the segment boundaries; the distance between two primary activators belonging to two different segments may be as small as one sentence and as large as a few paragraphs. Secondary activators also facilitate filtering out the irrelevant portions of text.

Form of an activator: can be *definite*, *indefinite* or *neither*, and is defined by a small number of rules. The notion of the *definite form* of an activator approximates computing anaphoric references to the entities introduced explicitly earlier in the text. For noun phrases, this form often coincides with a definite anaphora accomplished via noun phrases with the determiner *the*[1]. Examples of the definite forms of primary activators in the *terrorist attacks* domain are:

1. *the brutal murder of the*
2. *such a venture*
3. *attacks which*
4. *their attacks.*

The notion of the *indefinite form* of an activator is intended to capture a newly introduced entity. Examples of the indefinite forms of activators in the same domain are:

1. *another bomb*
2. *the attack on a bank*
3. *also killed.*

The *neither form* of an activator is intended to signal a total uncertainty (don't care condition) as to whether the entity is a new or old one. An example of the neither form is *exploded*. The forms of the primary activators for the sample stories in Figures 5 and 6 are the expressions enclosed by the small boxes.

Cue phrase: is a simple expression known to mark changes in some discourse structures (Schiffrin 1987). Examples of cue phrases are:

1. *meanwhile*
2. *in addition.*

Entity type: is a symbol signifying an entity type. Primary activators and segments are typed. A single text fragment may have multiple types. This happens, for example, when a single sentence mentions several entities of different types. Type (in)consistency is used to decide the presence of segment boundary. For the domains considered, there are a very small number of types, and additionally, most types are mutually disjoint. For the *terrorist attacks* domain, the types included: the most general type *attack* and less than ten subtypes such as *arson* and *bombing*. The *joint venture* domain had *venture* as its most general type and only one subtype *joint venture*.

[1] Bare pronoun anaphoric expressions such as *it* or *they* are ignored because they are not indicative of the domain-interest entities; noun phrases starting with a personal pronouns, e.g. *their attacks*, on the other hand, are assigned a form.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

3.3 Checks

The outcomes of the following checks influences the decision as to whether a subsequent paragraph or sentence continues describing an entity or starts describing a new entity:

Type consistency check verifies whether the type of the current primary activator is different from the type of the current segment; for example, the activator *girl* has a different type than the activator *boy*; some types are compatible with other types: the type *attack* is consistent with the types *robbery* and *murder*; This check is also used to decide if two consecutive segments should be merged. For example, they will be merged if their types are compatible and if the first primary activator in the second segment is in the *definite form.*

Definiteness of a primary activator check verifies the form of the primary activator;

Number check verifies the number of the primary activator; this number often, but not always, coincides with the syntactic number such as *singular* and *plural*;

Cue phrases check verifies the presence of cue phrases;

Filtering irrelevant text check, if requested, verifies the amount of text with no activators separating the current and previous primary activators.

---

if the current segment is a multi type segment
  then if the primary activator is in the same sentence as the last significant primary activator
    then the answer is no
    else if the type of the current primary activator is a member of the type of the current segment or it is of the most general type,
      current primary activator and last significant primary activator are either within the same paragraph, separated by one
      paragraph with a single sentence or are separated by one paragraph with at least one secondary activator,
      and the current primary activator is in definite form
      then the answer is no
      else the answer is yes
  else if the current primary activator is a secondary activator for the type of the current segment or (the last significant primary activator is
    a secondary activator for the current primary activator type and the current segment type is the most general type)
    then if the current primary activator and the last primary activator are separated by a paragraph with no secondary activators
      then the answer is yes
      else if the current primary activator and the last primary activator are within the same sentence
        then the answer is no
        else if the current primary activator is in indefinite form and it satisfies extra constraints
          then the answer is yes
          else the answer is no
    else if the last significant primary activator is a secondary activator for the current primary activator type
      then if the last significant primary activator and the current primary activator are within the same paragraph
        then the answer is no
        else the answer is yes
      else if the current primary activator type is inconsistent with the current segment type
        then if the current primary activator and the last significant primary activator are within the same sentence
          then the answer is no
          else then the answer is yes
        else if the type of the current segment is the same as the type of the current primary activator
          then if the current primary activator and the last primary activator are separated by a paragraph with no
            secondary activators
            then the answer is yes
            else if the current primary activator is in indefinite form and it satisfies extra constraints
              then if the current primary activator and the last primary activator are within the same sentence
                then the answer is no
                else the answer is yes
          else if the type of the current segment is the most general type
            then if the current primary activator and the last primary activator are separated by
              a paragraph with no secondary activators
              then the answer is yes
              else the answer is no
            else if the current primary activator type is the most general type
              then if the current primary activator and the last primary activator are separated by a paragraph
                with no secondary activators
                then the answer is yes
                else if the current primary activator and the last primary activator are within the same sentence
                  then the answer is no
                  else if the current primary activator is in indefinite form and satisfies extra constraints
                    then the answer is yes
                    else the answer is no

Fig. 3. Checking whether a new segment starts.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

The algorithm for computing the E- and SI-structures of the input text is shown below.

Input:
Text with identified sentence and paragraph boundaries, primary and secondary activators, and cue phrases

Output:
A set of possibly linked segments each of which contains relevant fragments of the story that describe entities of interest

Step 1: Identify basic segments

Consider subsequent primary activator:

1. **if** a new segment starts
   then compute the boundaries of the current and the new segment and adjust their activators
2. **if** the current segment continues
   then update the list of activators

if there are no more activators
then go to **Step 2**
else repeat **Step 1** for the next activator of the story

Step 2: Compute links among segments

Step 2.1: Check if E-structure present

Find three consecutive segments S_1, S_2, and S_3 such that

1. Types of S_1 and S_2 are not consistent
2. Types of S_1 and S_3 are equal
3. The form of the first primary activator of S_3 is definite

if the segments S_1, S_2, and S_3 satisfy the requirements
then E-structure is identified; go to **Step 3**
else no E-structure is identified; go to **Step 2.2**

Step 2.2: Check if SI-structure present

Step 2.2.1: Find SIB and SIA segments
SIB is a segment such that:

1. The form of its first primary activator is either indefinite and plural or neither and singular
2. The form of all the other primary activators is definite

SIA is a segment such that:

1. The form of its first primary activator is definite and plural
2. The form of all the other primary activators is definite

If the SIB or SIA segment is found
then go to **Step 2.2.2**
else no IS-structure is identified; go to **Step 3**

Step 2.2.2: Find individual segments
Find segments following SIB (preceding SIA) such that the type of each segment is smaller than or equal to the type of SIB (SIA);
if such segments are found
then full IS-structure is identified; go to **Step 3**
else partial IS-structure is identified; go to **Step 3**

Step 3: Output discourse structure of the input text

Algorithm: Computing discourse structure of a text: an algorithm

4 Examples

Figures 5 and 6 show structures computed by the discourse processing module implementing the algorithm for Sample Texts 1 and 2. The *joint venture* sample text exhibits the SI-structure, and the *terrorist attacks* sample text the E-structure. I will discuss the steps of the algorithm for the latter text in greater detail.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

Compute:
LPA – beginning of the paragraph with the current primary activator
LSA – beginning of the sentence with the current primary activator
RCT – end of the paragraph with the last primary activator of the current segment
if RCT is smaller than LPA
  then check secondary activators:
    if the paragraph following the paragraph with the last primary activator of the current segment contains at least one secondary activator
      then the current segment ends with the last word of the adjacent paragraph;
      else the current segment ends at RCT
    if the paragraph preceding the paragraph with the current primary activator contains at least one secondary activator
      then the new segment starts with the first word of the preceding paragraph
      else the new segment starts at LPA;
  else current segment ends at index (LSA - 1), and new segment starts at index LSA

Fig. 4. Computing the enclosing boundaries of the current and new segments.

<DOCNO> 0063 </DOCNO>
<DD> JANUARY 8, 1991, TUESDAY </DD>
<SO> Copyright (c) 1991 Kyodo News Service </SO>
<TXT>

;; ********** *Segment nr= 1, SIB, Type= joint venture* **********

NISSEI BUILD KOGYO CO. SAID TUESDAY IT WILL LAUNCH [JOINT VENTURES] IN ITALY, CANADA, SOUTH KOREA, AND TAIWAN TO CONSTRUCT HIGH-RISE PARKING FACILITIES. [THE KANAZAWA-BASED COMPANY] SAID [THE VENTURES] WILL BE EQUALLY OWNED, WITH NISSEI PROVIDING ELEVATORS, CONTROL SYSTEMS, AND TURNTABLES. LOCAL PARTNERS WILL HANDLE PRODUCTION OF STEEL FRAMES, ASSEMBLY, AND MAINTENANCE.

;; ********** *Segment nr= 2 Type= joint venture* **********

[THE COMPANY] SAID [ITS JOINT VENTURE] IN CANADA PLANS TO EXPORT FACILITIES TO SAN FRANCISCO AND OTHER CITIES ON THE U.S. WEST COAST.

;; ********** *Segment nr= 3 Type= joint venture* **********

[THE COMPANY] SAID [THE NEW COMPANIES] TO BE ESTABLISHED WILL BE IMER NISSEI AND IMER INTERNATIONAL IN POGGIBONSI, ITALY.

;; ********** *Segment nr= 4 Type= joint venture* **********

[THE OTHERS] WILL BE THE BALSHINE FOUNDATION AND GEORGE THE THIRD AND SON IN VANCOUVER, CANADA= SAMICK NISSEI CO. AND WOOSUNG HEAVY INDUSTRIAL CO. IN INCHON, SOUTH KOREA= AND TAIWAN NISSEI, WHICH WILL BE FORMED BY NISSEI AND A GROUP OF MEDIUM AND SMALL ENTERPRISES IN TAIWAN.
</TXT>

</doc>

Fig. 5. The sample text in the *joint venture* domain exhibits the SI-structure. Information from the SIB segment, number 1, should be pairwise combined with the information in each segment 2, 3 and 4.

Algorithm steps for terrorist attacks sample text

In the first step, the program identifies basic segments. The primary activators *dynamite*, *charge* and *exploded* signal a *bombing* event. The definite form of the next two primary activators *explosion* makes the program decide that the subsequent sentences describe this event. The next paragraph contains no primary or secondary activators, and if the filtering were on, it would have been judged as irrelevant and would have resulted in the first text fragment.

The next primary activator is *charge*. Its type is consistent with the type of the current segment and its form is definite – the current segment continues. The next

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

primary activator *attack* points to an event of type *attack* consistent with the current segment and the form of the activator is definite – the current segment continues.

Two subsequent primary activators are *attack* and *killing*. The form of the first activator is indefinite – the module starts a new *attack* segment. The second activator points to a *murder* event and is in the indefinite form – the module changes the type of the segment to *murder*.

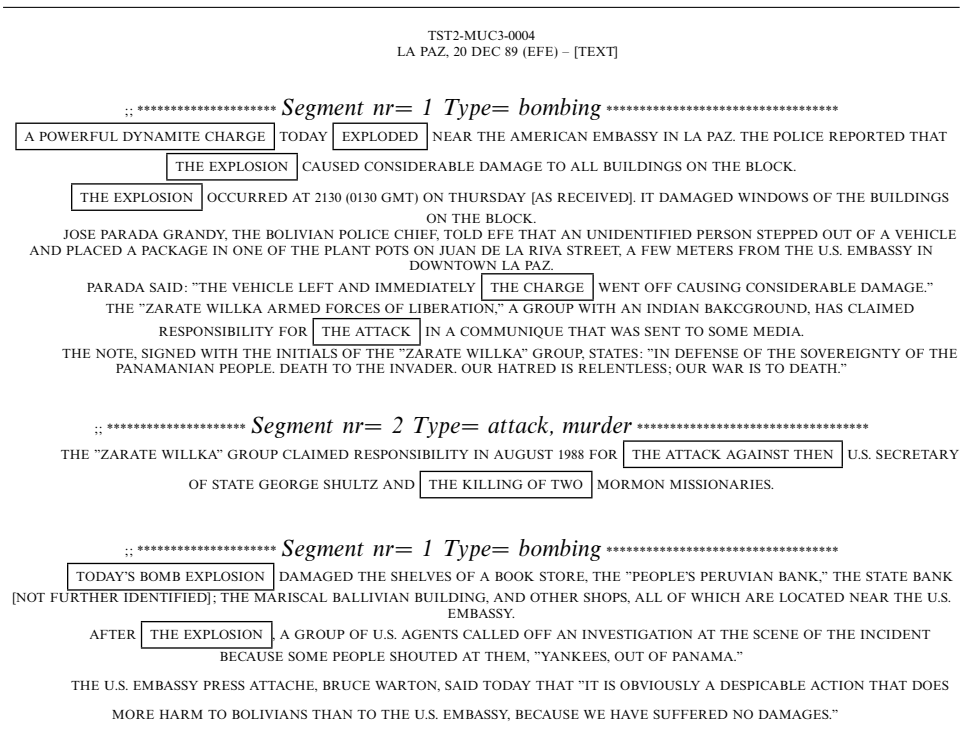


Fig. 6. The sample text in the *terrorist attacks* domain exhibits the E-structure. The segment number 1 encloses the segment number 2. Information from two nonconsecutive fragments of the segment number 1 must be combined.

The next paragraph contains two primary activators *bomb* and *explosion*. The type *bombing* is different from the type *murder* – the module creates a new *bombing* segment. The next paragraph contains the activator *explosion* in the definite form – the segment continues. The last paragraph contains a secondary activator *damages* and is also included in this segment.

In the second step, the module links the first and the third segments into a single segment because these segments are of the same type. The type of the first segment is inconsistent with the type of the second segment, and the first primary activator of the third segment is in the definite form.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

5 Discussion

5.1 Heuristics-based good performance

The algorithm presented for computing the SI- and E-discourses is heuristic in nature and has its weaknesses. It does not always recognize multiple entities; for example, when they are described in a single clause. In certain cases, the algorithm incorrectly produces two segments for a single entity. This typically happens when filtering out irrelevant text is necessary, and almost never happens otherwise.

Despite its heuristic nature, the algorithm performs quite well, an estimated earlier 85–95%, and correctly computes text segments and their linking in a large corpus of real linguistic data with a good mix of articles that exhibit no structure, articles with full and partial SI-structures and E-structures, and articles that are relevant and irrelevant for a given domain.

5.2 Good predictors of segment boundaries

It was surprising that some very simple heuristics produced such reliable results. One such heuristics for starting a new segment, i.e. for identifying a segment boundary, is the change of the syntactic number of an activator. A simple switch from (to) *singular* to (from) *plural*, or *vice versa*, is a very strong indicator of a segment boundary.

Another strong boundary predictor is based on the notion of (in)definiteness. This heuristics is apparently a pretty good approximation of telling a new entity from the one introduced earlier in the text. The fact that such known devices as anaphora resolution would be relevant here is no surprise. What is surprising is the fact that computing certain classes of definite anaphors can be accomplished with very little and very local parsing and other so-called in-depth processing, hopelessly beyond any NLP system on any scale other than processing one or two stories.

5.3 Poor predictors of segment boundaries

Some other seemingly reasonable heuristics were found surprisingly unreliable. For example, starting a new segment based on the presence of a cue phrase. While a cue phrase always marks some boundary, it rarely corresponds to the boundary of interest here, and consequently introduces a high margin of errors.

Another surprise was the fact that paragraph boundaries do not coincide well with the segment boundaries. This is in part because of the (in)completeness of knowledge about different entities as well as their author-perceived importance varies. For describing some entities, writers need several paragraphs, while for others, what is known can be expressed in a single sentence. Another reason for the observed poor alignment of the paragraph and segment boundaries is the fact that some entities are viewed as more important than others and more space is allotted to them.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

5.4 Why not both?

It is interesting that stories in the corpora considered, frequently exhibit these structures, either the SI- or the E-structures, but never both. It may have something to do with their length. One–two page articles are still relatively short.

5.5 More powerful parser?

The performance of the algorithm can be improved in a number of ways. One obvious way is to employ a more powerful parser capable of parsing languages that go beyond regular languages. This would improve somewhat the correctness of the identifying definite and indefinite forms of the activators. The problem is that the complexity of English combined with the large number of stories is a prohibitive factor: most, if not all, state-of-the-art parsers are not capable of a high-precision correct parsing (Sundheim 1991). The size of the corpora makes the processing time an important factor, and one problem with powerful parsers is that they are extremely slow.

5.6 Better algorithm for definite anaphora resolution?

Another obvious way to try to improve the performance of the algorithm is by employing a powerful definite anaphora resolution algorithm. This again largely hinges upon the unrealistic for the considered corpora assumption about correct sentential-level parsing and in-depth processing.

5.7 Further soften domain-dependency

Activators reflect both the specificity of the domain as well as information needed to be extracted from texts. Producing reasonable activators is a very hard, and is extremely time- and money-consuming effort. My guess is that the only reasonable way of eliminating hand-produced activators are machine learning inductive methods for extracting such activators from marked corpora. It is not clear to me at all that this can indeed be done.

One thing that would be interesting to explore is to quantitatively assess how various heuristics, the strongly domain-dependent and the losely-or-not-all domain-dependent, contribute to the overall performance. For some texts, the domain-independent components are enough to recover the underlying structures. This was largely the case for the *joint venture* domain.

5.8 Heuristics are so hard to beat!

It has been my experience that unless in-depth processing components such as powerful parsers or sophisticated definite anaphora resolution are close to perfect, it is almost impossible to beat the heuristics incorporated in my algorithm, and certainly it is not possible to do it on a large corpus. In fact, using such slightly

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

less than perfect 'improvements' often significantly degrades the overall system performance.

The *pear story* (see Figure 7) brings this point home. It contains some very tricky definite anaphoras, which did not turn problematic for the heuristics embedded in my algorithm. Below, the segmentation of the *pear story* produced independently by four human subjects in the experiments (Litman and Passonneau 1993), and the structure produced by the new algorithm, are shown.

The following discourse structures (segmentations) were produced for the *pear story*; the lines starting with 'BOUNDARY people:' indicate boundaries produced by people participating in the experiments of Litman and Passonneau (1993); the lines starting with 'BOUNDARY: DPM' indicate boundaries produced by the algorithm presented here:

1.1 Okay,
1.2 [2.05.. u-h [1.1]] the movie is basically about uh [.2] u-m [.85] a number of [.45] individuals,
BOUNDARY DPM: segment 1 ends

1.3 [.6] uh a guy who's picking pears,
BOUNDARY DPM: segment 2
1.4 [2.1 [1.0] u-m [.6]] and a kid on a bicycle.
BOUNDARY people: 2 subjects (c@, d@)
BOUNDARY DPM: segment 3 ends

2.1 Basically those are the two.. protagonists in this.
BOUNDARY people: 2 subjects (b@, c@)
BOUNDARY DPM: segment 4 ends

3.1 [2.8 [1.05] And.. um [.6]] the guy who is picking pears,
3.2 [3.15 um [2.35] um [.35]] picks the pears and puts them in a [.45] in um [.4] these baskets that he has.

4.1 [1.6 [.9] U-h..] and he's picking the pears,
BOUNDARY people: 3 subjects (a@, b@, c@)
BOUNDARY DPM: segment 5 ends

4.2 [1.0 and [.45] and um..] along comes a man with a donkey.

5.1 Uh uh a don uh a goat.

6.1 [.45] And he comes along.. by.. you know,
6.2 [.4] passes him.
BOUNDARY people: 4 subjects (a@, b@, c@, d@)
BOUNDARY DPM: segment 6 ends

7.1 [.4] And then this kid comes along with a bicycle.

8.1 [.55] And he rips off.. one of the [1.0] baskets of.. of pears that he has.
BOUNDARY people: 1 subjects (b@)

9.1 [.8] So the ki the the um.. the boy goes along,
9.2 and he has [1.4.. um [.8]] he's riding his bicycle,
BOUNDARY people: 1 subjects (c@)
BOUNDARY DPM: segment 7 ends
9.3 a-nd he looks at.. a girl,
9.4 that was coming the other way,
9.5 riding a bicycle,
BOUNDARY people: 2 subjects (b@, c@)
9.6 [.9 [.7] uh] he loses his hat,
9.7 a-nd.. there's a stone in the way,
9.8 so his bicycle falls over,
9.9 .. and the pears get [3.5 [l.65] um [.85] um [.4]]
fall down on the ground.
BOUNDARY people: 4 subjects (a@, b@, c@, d@)
BOUNDARY DPM: segment 8 ends

10.1 [.75 [.. U-m [.5]] there's some kids,

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

BOUNDARY DPM: segment 9 ends

10.2 there are three other boys,
10.3 .. who are there.
BOUNDARY people: 1 subjects (b@)

11.1 [.35] They help him,
11.2 .. get straightened out,
11.3 .. put the pears back,
11.4 [.55] in the basket,
11.5 straighten out his bicycle,
11.6 and so forth.
BOUNDARY people: 3 subjects (b@, c@, d@)
BOUNDARY DPM: segment 10 ends

12.1 [.85] And he goes on his merry way.
BOUNDARY people: 1 subjects (a@)
BOUNDARY DPM: segment 11 ends

13.1 [1.25 But then.. um [.35]] the boys realize that he's forgotten his hat.
BOUNDARY DPM: segment 12 ends

14.1 [.55] So one of the boys whistles to him,
14.2 [.45+ and [.45]] stops him,
14.3 [.2+ and [.2]] gives him his hat back.
BOUNDARY people: 3 subjects (a@, b@, c@)
BOUNDARY DPM: segment 13 ends

15.1 [3.55 [1.5] And then.. um [1.1]]the boy with the pears.. gives [.55] the boy who just gave him his hat [1.85 [.7] um [. 75]] three pears to.. divide among his friends.
BOUNDARY DPM: segment 14 ends

16.1 [1.9 [l0] And.. then..] the boys go.. um.. walking along,
16.2 eating their pears.
BOUNDARY people: 4 subjects (a@, b@, c@, d@)
BOUNDARY DPM: segment 15 ends
um [l.0]] then the man [.25] uh.. who was picking pears,
17.2 [.6] comes down 17.1 [2.9 [1.1] And from his- [2.45 [.35] um [1.65]] his.. ladder,
17.3 [.65] where he's been picking these pears,
17.4 [.5] and he's goes to empty out the ones that he's just picked.

18.1 [.6] And he notices that [.35] instead of the three baskets that he had before,
18.2 [.4] there are only two.
BOUNDARY people: 1 subjects (b@)

19.1 [.8] And.. so.. he's puzzled,
BOUNDARY people: 1 subjects (a@)
19.2 [.45] and.. just when he realizes that [.45] one basket is.. gone,
BOUNDARY DPM: segment 16 ends

19.3 [.45] the three boys come along,
19.4 [.8] eating their pears.
BOUNDARY people: 3 subjects (b@, c@, d@)
BOUNDARY DPM: segment 17 ends

20.1 [.75 And [.65]] you're left with this.. dilemma,
20.2 [.4] what does this guy laugh you know what does this guy really think.
BOUNDARY DPM: segment 18 ends

21.1 I guess he thinks that [.95] "I wonder if those guys ripped off with my pears or what." He just doesn't know.
BOUNDARY DPM: segment 19 ends

22.1 [.35] He was up in the tree when.. the boy on the bicycle ripped off the uh /?s/.. the pears.

23.1 Okay?
BOUNDARY people: all subjects (a@, b@, c@, d@)
BOUNDARY DPM: segment 20 ends

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

```
Okay,
uh  the movie is basically about uh um a number of individuals,
uh a guy who's picking pears,
um and a kid on a bicycle.
Basically those are the two protagonists in this.
And um the guy who is picking pears,
um um picks the pears and puts them in a in um these baskets that he has.
Uh and he's picking the pears,
and and um along comes a man with a donkey.
Uh uh a don- uh a goat.
And he comes along by you know,
passes him.
And then this kid comes along with a bicycle.
And he rips off one of the basket sof of pears that he has.
So the ki- the the um the boy goes along,
and he has  um he's riding his bicycle,
and he looks at a girl,
that was coming the other way,
riding a bicycle,
uh he loses his hat,
and  here's a stone in the way,
so his bicycle falls over,
and the pears get um um fall down on the ground.
Um there's some kids,
there are three other boys,
who are there.
They help him,
get straightened out,
put the pears back,
in the basket,
straighten out his bicycle,
and so forth.
And he goes on his merry way.
But then  um  the boys realize that he's forgotten his hat.
So one of the boys whistles to him,
and stops him,
and  gives him his hat back.
And then um the boy with the pears gives
the boy who just gave him his hat um
three pears to divide among his friends.
And then the boys go um walking along,
eating their pears.
And um then the man uh who was picking pears,
comes down from his um his ladder,
where he's been picking these pears,
and he's goes to empty out the ones that he's just picked.
And he notices that instead of the three baskets that he had before,
there are only two.
And so he's puzzled,
and just when he realizes that one basket is gone,
the three boys come along,
eating their pears.
And  you're left with this dilemma,
what does this guy \{laugh\} you know what does this guy really think.
I guess he thinks that  "I wonder if those guys ripped off
with my pears or what."
He just doesn't know.
He was up in the tree when the boy on the bicycle ripped off
the uh the pears.
Okay?
```

Fig. 7. The pear story: a celebrated story studied for discourse phenomena.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

5.9 Generality

Extension to other domains and other genre of texts remains to be further investigated. The fact that the approach generalized easily to the *joint venture* domain and that it produced intuitive results for the very complex pear narrative in an entirely different domain jointly constitute a strong positive evidence in its favor.

Additionally, it is very plausible to assume that in any domain and any genre of text, some entities and their various groupings will be talked about. The specificity of a domain will most probably require reasoning about some special properties facilitating the process of identifying and distinguishing different entities.

6 Conclusions

Two types of discourse structure common in factual reports have been identified. An algorithm for computing these structures in a large corpus of real linguistic data on English has been presented. The algorithm is based on phenomena observed in two different domains. Computing these discourse structures facilitates the task of creating a text-level representation of input text and the task of correctly propagating information about the entities of interest distributed nonlinearly throughout the text.

Acknowledgements

This work was done at the Artificial Intelligence Laboratory, GE Research & Development Center, and at the Department of Computer Science, Wayne State University. I would like to thank Dale Russell for his help with data verification, testing the reliabililty of some of the heuristics and constructive comments on the approach. At the time, Dale was a PhD student in the Department of Computer Science, University of Illinois at Urbana-Champaign, and a GE Summer 1992 student working with me.

References

MUC (1992) *Proceedings of MUC-4, Fourth Message Understanding Conference*. Morgan Kaufmann.

Chafe, W. L. (ed.) (1980) *Advances in Discourse Processes: Vol. III*. Ablex.

Grosz, B. and Sidner, C. (1986) Attention, intensions and the structure of discourse. *Computational Linguistics*, **12**(3).

Iwańska, Ł. (1991) Discourse Processing Module. *Technical Report 91CRD180*, Artificial Intelligence Laboratory, GE Research & Development Center.

Iwańska, Ł., Appelt, D., Ayuso, D., Dahlgren, K., Stalls, B. G., Grishman, R., Krupka, G., Montgomery, C. and Riloff, E. (1991) Computational Aspects of Discourse in the Context of MUC-3. *Proceedings of the Third Message Understanding Conference, MUC-3*, pp. 256–282. Morgan Kaufmann.

Litman, D. and Passonneau, R. (1993) Empirical investigations of discourse segmentation. *31-st Annual Meeting of the Association of Computational Linguistics*.

Mann, W. and Thompson, S. (1991) Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, **8**(3): 243–281.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press

Morris, J. and Hirst, G. (1991) Lexical cohesion computed by thesaural relations as as indicator of the structure of text. *Computational Linguistics*, **17**(1).

Moser, M. and Moore, J. (1996) Toward a Synthesis of Two Accounts of Discourse Structure. *Computational Linguistics*, **22**(3): 410–420.

Schiffrin, D. (1987) *Discourse Markers. Studies in International Sociolinguistics: No. 5*. Cambridge University Press.

Sundheim, B. (ed.) (1991) *Proceedings of MUC-3, Third Message Understanding Conference*. Morgan Kaufmann.

van Dijk, T. A. and Kintsch, W. (1983) *Strategies of Discourse Comprehension*. Academic Press.

https://doi.org/10.1017/S1351324997001770 Published online by Cambridge University Press