A Discourse Grammar for Processing Arguments in Context

Juyeon KANG [a,1], Patrick SAINT-DIZIER [b]

[a] *Prometil, Toulouse, France*

[b] *IRIT-CNRS, Toulouse, France*

Abstract. In this paper, we develop a linguistic and conceptual analysis of argument coumpounds, i.e. groups of closely related arguments and their related contexts expressed by means of discourse relations. An implementation in Dislog is presented with an indicative evaluation of the results.

Keywords. Natural language processing, discourse, conceptual representation.

Introduction

Arguments in written texts or dialogues seldom come in isolation, as independent statements. They are often embedded into a context that indicates e.g. circumstances, restrictions, purposes, and various forms of elaborations. Relations between an argument and its context may be conceptually complex. Furthermore, corpus analysis shows that, besides their contexts, arguments often appear in small and closely related groups or clusters where they share similar aims, where the first argument is complemented, supported, reformulated or elaborated by the subsequent ones. These small groups may also include statements in contrastive or concessive configurations.

In terms of language realization, clusters of arguments and their related context may be all included into a single sentence via coordination or subordination or may appear as separate sentences. In both cases, the relations between the different elements of a cluster are realized by means of conjunctions, connectors, various forms of references and punctuation. We call such a cluster an **argument compound**. The claim, behind this term, is that the elements in a compound form a single, possibly complex, unit, which must be considered as a whole from a conceptual and argumentative point of view.

Language expressions of arguments are often very diverse and complex, making their automatic identification in texts a very challenging task. Besides language complexity, a large number of arguments are not clearly marked by specific linguistic cues, therefore, it is often necessary to have recourse to semantics and pragmatics to identify, delimit and understand them and then identify the relations within compounds.

Technical documents (e.g. procedures, product manuals, specifications) form a linguistic genre with restricted linguistic constraints in terms of lexical realizations, including business or domain dependent aspects, grammar, style and overall organization.

[1] Corresponding Authors: j.kang@prometil.com, stdizier@irit.fr

These documents are designed to be as efficient and unambiguous as possible, this make their automatic analysis less problematic. For that purpose, they tend to follow relatively precise authoring principles concerning both their form and contents. Technical documents abound in various classes of arguments, in particular recommendations, warnings, advice, requirements and regulations. These arguments appear in isolation, often with several supports, except for requirements. Automatically identifying argument compounds in technical texts and producing a conceptual representation adequate for subsequent treatments is the major concern of this paper. For that purpose, we develop a discourse grammar that accounts for the conceptual structure of argument compounds. The model is based on logic, logic programming and constraint satisfaction; it is implemented on the <TextCoop> platform via the Dislog language.

This short paper further elaborates on results presented (1) in [10], where processing isolated warnings and advice is presented together with their implementation in Dislog, (2) in [15], where we show that discourse structures, for which a detailed semantic analysis is developed, can be interpreted as argument supports in opinion analysis and (3) in [6] dedicated to requirement mining. The innovative aspects of this paper concern the linguistic analysis of arguments compounds that takes into account the role of discourse relations, and a few elements of a conceptual model that accounts for the relations between the constituents of a compound.

1. Linguistic and Conceptual Analysis

The linguistic structure of arguments as isolated utterances or as networks of arguments has been investigated in a number of works in linguistics and cognitive semantics, e.g. [16]. Much less has been developed from a technical perspective in computational linguistics, e.g. [9]. Difficulties come from the large diversity arguments may have in language, the need of contextual information to identify them and the difficulty to relate arguments with their supports or with other arguments, in particular when they are not adjacent in a text or a dialogue.

In terms of discourse, the RST [7], [14] has been very influential and had consequences on works such as [4], [12], and [17]. Several approaches, based on corpus analysis with a strong linguistic basis, are of much interest for our approach. Relations have been investigated together with their linguistic markers in e.g. [3], [8]. [13] developed a useful typology of markers.

Our approach merges argument and discourse structure analysis. The typical configuration of an argument compound can be summarized as follows:

```
CIRCUMSTANCE(S) / CONDITION(S), PURPOSE(S) -->
    [ARGUMENT CONCLUSION + SUPPORT(S)]*
        <-- PURPOSE(S), CONCESSION(S) / CONTRAST(S), ELABORATION(S)
```

The kernel of this structure is the organized set (noted with the *) of arguments and their supports. The main argument occurs in general first, it is then followed by secondary arguments. The compound starts with circumstances and conditions, possibly purposes, when they have a wide scope. Then follows the set of arguments and their supports. The compound ends by purposes, concessions or contrasts and elaborations.

At the language realization level, this logical organization may not be realized straightforwardly. In particular, we observed that:
- the initial group, that should logically precede the set of arguments, may be inserted between arguments,
- the last group, that should also logically follow the set of arguments, may be inserted between these arguments,
- purposes may be realized as supports or vice versa,
- an argument may have several supports, possibly with different orientations, supports may not be adjacent to their related conclusion,
- supports may be inserted within their conclusion, instead of following or preceding it.

Here are a few argument compounds, where a few tags we have defined for this task have been inserted:

Ex. 1. <ArgCompound> <purpose> Cleaning your leathers.< /purpose> <mainArg> < conclusion> Prefer natural products. < /conclusion> <support polarity="-"> They are more expensive < /mainArg> but <support polarity ="+"> they will have a longer effect and make minor repairs. < /support>< /advice> </ArgCompound>

Ex. 2. <ArgCompound> <definition> Inventory of qualifications refers to norm YY. < /definition> <mainArg> <conclusion> Periodically, an inventory of supplier's qualifications shall be produced. < /conclusion> < /mainArg> <secondaryArg> <conclusion> In addition, the supplier's quality department shall periodically conduct a monitoring audit program. < /conclusion> < /secondaryArg> <elaboration> At any time, the supplier should be able to provide evidences that EC qualification is maintained. </elaboration> < /ArgCompound>

Ex. 1 illustrates the case where an argument of type advice has several supports with different orientations. The contrastive connector *but* introduces the inversion of the polarity. The first support is not really an attack, but a kind of contrast. Ex. 2 is a requirement compound. It shows how a definition makes the requirements more accurate. A secondary requirement complements the main one, which is further elaborated in the last sentence. This sentence is not a requirement because of the modal *should be able to* which is not injunctive.

Our corpus contains a variety of documents. It is composed of 228 texts where 154 texts have been selected for the development corpus, and 74 texts have been selected to make an indicative evaluation of the results.

Typology of argumentation in technical documents:

Requirements and regulations, requirements [5] and regulations form a special class or arguments, with specific linguistic forms and a very injunctive orientation. Their support(s) must not be confused with purpose clauses: their role is to justify the requirement, its importance, and the potential risks and difficulties that may be encountered. Their identification is based on precise patterns in a sentence [6] such as: `[modal(shall, must, have to) + infinitive verb].` A comprehensive requirement is e.g. *an inspection shall be carried out monthly for a correct cleaning of the universal joint shafts.*

Prevention arguments or warnings basically explain and justify an instruction or a group of instructions. These are very frequent in most types of technical documents. Formulations with a negative polarity are frequent, their structure is given in [10]

Performing or advice arguments are less imperative than the previous ones, they express advices or task evaluations. These are also very frequent, in particular in documents designed for novices [11].

Threatening arguments are less frequent. The reader is directly involved in the consequences. These arguments have a strong impact on the users attention when he realizes the instruction.

2. Processing argument compounds

2.1. Identification and delimitation of argument compounds

In general, the development of relatedness criteria between an argument conclusion and its support(s) or related discourse structures is complex, in particular when there is no strict adjacency, which is frequent.

The principle is that all the statements in an argument compound must be clearly related either by the reference to a precise theme or via specific marks that define **cohesion links**. Such links must be made explicit in technical texts to avoid ambiguities. They can be identified and categorized in a relatively clear way. The theme of a compound is a nominal construction (object or event, non human, e.g. *inventory of qualifications* in Ex. 2)). Links between arguments and related discourse structures are defined on the following basis:
- the use of the theme in the sentences that follow or precede the main requirement (e.g. *inventory of qualifications* in Ex. 2). The theme can possibly undergo morphological variations, a different determination (e.g. *safety test, all safety tests*) and simple syntactic variations. This first situation occurs in about 70% of the cases. The theme is the subject of the clause (about 65% of the cases) when the subject does not denote a human actor or the direct object in the other cases (35%).
- the use of a pronoun referring to the theme, when the theme is in subject position in the sentence that precedes (about 25% of the cases),
- the use of a more generic term than the theme, used as a simple form of reference, e.g. *this process, this constraint*, in the utterances that follow the main requirement (about 20% of the cases). We identified 42 such general purpose terms.
- the use of discourse connectors to introduce a sentence, e.g. *however, for that purpose, if*, etc., found in about 35%
- the use of sentence binders such as: *for information, in this case, at any time, next, also,* etc. found in 20% of the cases.

These criteria may overlap, in particular in utterances other than requirements where the theme and a connector can be found.

2.2. Discourse Relations in a compound

In an argument compound, arguments and sentences other than arguments are linked by means of discourse relations. This defines a kind of network of relations. The relations between the main argument and the secondary ones are essentially contrasts, concessions and specializations. The structure and the markers and connectors typical of discourse relations found in technical texts are developed in [11]. These have been enhanced and adapted to the compound context via several sequences of tests on our corpus. The main relations found are the following:
- **contrast**, [19] and [12], is a relation between two arguments that introduce one or more

equivalent but alternative views, but which refer to a unique situation.
- **concession** states a general requirement followed by an apparently contradictory argument that could be admitted as an exception. The contradiction with the implicit conclusion which can be drawn from the first argument is partial (e.g. [2]). Concessions are often categorized as denied phenomenal causes or motivational causes.
- **specializations, and subsequent constraints**.
- **information and definitions** mainly occur before the main argument.
- **elaborations** follow an argument, they develop some of its facets to facilitate its understanding. Elaborations may play the role of supports. Since this relation is very underspecified, we consider it as the by-default relation in the compound. A categorization of the main functions covered by elaboration are in particular: localization, precision, focus, future actions, application domains, constraints, prerequisites.
- **illustration** provides related examples.
- **result** specifies the outcome of an action. Its linguistic structure is basically the active-inchoative alternation that describes the expected result, implemented via the use of the theme combined with the main verb past participle or with an aspectual verb denoting completion or quasi-completion.
- **circumstance** introduces a kind of local frame under which the argument compound is valid or relevant. Circumstances often appear before the argument(s) they apply to. Circumstances introduce temporal, spatial or factual contexts or particular events or occasions.
- **purpose** which expresses the underlying motivations of the argument compound. It must not be confused with argument supports.

3. Towards a conceptual model

The role of the conceptual model is to represent the relations between the various units of the compound in order to allow to draw inferences between compounds, to make generalizations and to check coherence between argument compounds and within compounds, e.g. [1], [19]. A preliminary model is briefly developed in this section.

The goal is to define in more formal and precise terms the semantics that each of the discourse relations *Rel* has w.r.t. to arguments. Such an investigation has been initiated for opinion mining for some of these relations [15].

Contrasts and concessions are particularly interesting in argumentation from an inferential point of view. Connectors such as *but, however, nevertheless* characterize these relations. [18] makes a first attempt at characterizing some of these connectives, focusing for the most part on *but*. [1] gives an account of a procedural analysis of 'however' and 'nevertheless' within the framework of Relevance Theory. From an inferential point of view, both contrast and concession relations have the following semantics:
$[P \Rightarrow (R1)\ 'but'\ Q \Rightarrow (R2)]$
where P and Q are contrasted and R1 and R2 are the inferences drawn from P and Q respectively. R1 and R2 are at least partly inconsistent. 'But' partly cancels the inferences R1 made from P. In the contrast relation, it is mainly the conclusions R2 which form the main conclusion of the structure, by partly denying R1. In the concession relation, R1 remains the main conclusion, but with some nuances borrowed from R2.

Concerning the consequence and result relations, current work is largely focused on their argumentative properties from a logical perspective. Consequence is often associ-

ated with the conclusion of an argumentation. Consequence is seen as a form of elaboration in [15], distinct from the final conclusion of a complex argumentation. In 'P therefore Q', P (or the inferences P_i made from P) somehow imply and logically leads to Q (or the inferences Q_j made from Q). Q keeps its own set of inferences Q_j, which are consistent with P_i. The elaboration is a form of composition of P_i and Q_j:
$[P \Rightarrow (P_i)\ 'therefore'\ Q \Rightarrow (P_i)]$.
The resulting inferences are: $Q \Rightarrow Q_j \cup P_i]$

4. Implementation in Dislog

The **TextCoop platform and the Dislog language** (for Discourse in Logic) have been primarily designed for argumentation and discourse processing [10]. TextCoop is based on Logic Programming, it is a platform that includes:
(1) **Dislog**, which is a logic-based language designed to describe in a declarative way discourse structures and the way they can be bound via selective binding rules,
(2) **an engine** associated with a set of processing strategies. Dislog rules are processed according to a cascade that specifies their execution order. This engine offers several mechanisms to deal with ambiguity and **concurrency** when different discourse structures can be recognized on a given text fragment,
(3) **a set of active constraints**, in the sense of Constraint Logic Programming, that state well-formedness typical language and of discourse structures (e.g. precedence, dominance, bounding nodes); these can be parameterized by the grammar writer,
(4) **input-output facilities** (XML, MS Word), and interfaces with other environments
(5) a set of **lexical resources** which are frequently used in discourse analysis (e.g. connectors),
(6) a set of about 180 **generic rules** that describe 12 frequently encountered discourse structures such as reformulation, illustration, cause, contrast, concession, etc.

Let us now breifly review the different steps of the parsing process.

Higher-order programming for compound delimitation: An argument compound is identified by its theme and by its boundaries following the criteria given above. Let T_0 be the theme identified in the main argument. Let $variant(T_0)$ be the finite set of the n variants which can be generated from T_0 via functions producing morphological variants, generalizations, etc.:
$variant(T_0)\ = \{T_i,\ i \in [1,n]\}$.
Then all the sentences S_j, $j \in [1,k]$, $k < 6$ (assuming that a compound has a maximum of 5 sentences, but this is a parameter) in the compound must meet the following constraints, expressed by the following two Dislog rules:
$\forall S_j\ j \in [1,k]\ \exists\ i,\ i \in [1,n]$*; sentence* $\rightarrow$ *T_i, gap(G), eos. / gap(G1), verb, T_i, gap(G2), eos.*
These rules introduce a kind of higher-order programming with a quantification on the form of the theme. The rules state that the theme or its variants must appear in the set of related sentences either in subject of object position.

Selective binding rules: Selective binding rules are essentially based on already tagged structures. Binding rules allow, under constraints, to bind arguments together or with discourse structures, under constraints. Binding rules are abstract, higher-order schemas. For example, binding the main argument R1 with the structure G1 (e.g. G1 = definition) that precedes it is represented as follows:

```
<X>, gap(G1), </X> <req status="main"> gap(R) </req> → <Argcompound>
<X>, G1, </X> <req status="main">, R, </req> </Argcompound>.
```

This rule inserts G1 and R into the <Argcompound> tag. X is a variable that stand for any structure provided that the precedence constraints (see next subsection) are met.

Constraints: Constraints mainly encode in this investigation precedence constraints (" < " encodes precedence). In Dislog, they are specified in a very declarative manner. The TextCoop engine checks that these constraints are met at each step of the processing. The following constraints induce a partial ordering of arguments and discourse structures in an argument compound:
information, definition < *argument.* : Information and definition occur before any argument.
Argument < *result.* : result always follows the relevant arguments.
Argument < *concession, contrast.* : Contrasts and Concessions always follow the arguments on which they operate.
contrast <> *concession.*: Contrasts and concessions cannot co-occur in a compound.

Finally, an argument compound must be fully realized within a paragraph, therefore, the node paragraph is a bounding node: *bounding_node([paragraph])* for binding rules.

The system designed for compound analysis is very declarative. It is composed of a set of rule clusters, associated lexical entries, and constraints. Rule clusters are activated one after the other with an order specified in a cascade. This cascade allows, among others, to specify priorities (a cluster must be fully processed before another one is activated) and to avoid ambiguities.

4.1. Indicative evaluation

The following preliminary indicative evaluation is designed to identify improvement directions. The evaluation has been realized on our test corpus where we have manually annotated 200 argument compounds. Compound identification by the system produces the following results:

	precision	recall
identification	93%	88%
opening boundary	96%	91%
closing boundary	92%	82%

The closing boundary is more difficult to identify because some terms out of the compound can be interpreted as theme variants. The identification of discourse structures in a compound produces the following results:

relation	nb of rules	nb of annotated structures	precision	recall
contrast	14	24	84	88
concession	11	44	89	88
specialization	5	37	72	71
information	6	23	86	80
definition	9	69	87	78
elaboration	13	107	84	82
illustration	20	42	91	83
result	14	97	86	80
circumstance	15	102	89	83
purpose	17	93	91	83

Some relations have more elaborated sets of rules because they have been reused and improved from previous experiments. Information and definition are not necessarily identified on the basis of marks but on their position in the compound, which is also a vague criterion. In general, however, results are good for discourse analysis.

5. Perspectives

In this paper, we have developed a linguistic model for the analysis and the representation of argument compounds. This contribution illustrates and investigates the complexity of argument constructions and the development of a conceptual model. Our results form a kind a discourse grammar dedicated to argument compounds. The specific discourse relations we have identified are conceptually characterized, with the functions they play, so that inferences can be drawn within and between argument compounds.

References

[1] D. Blakemore, Relevance and linguistic meaning : the semantics and pragmatics of discourse markers, CUP, Cambridge, (2002).

[2] E. Couper-Kuhlen, B. Kortmann, *Cause, Condition, Concession, Contrast: Cognitive and Discourse Perspectives*, Topics in English Linguistics, No 33, de Gryuter (2000).

[3] J. Delin, A. Hartley, C. Paris, D. Scott, K Vander Linden, Expressing Procedural Relationships in Multilingual Instructions, Proceedings of the Seventh IWNLG, (1994), 61-70.

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

[5] E. Hull, K. Jackson, J. Dick, *Requirements Engineering*, Springer, (2011).

[6] J. Kang, P. Saint-Dizier, Discourse Structure Analysis for Requirement Mining, International Journal of Knowledge Content Development and Technology, Vol 3(2), (2013).

[7] W. Mann, S. Thompson, *Rhetorical Structure Theory: Towards a Functional Theory of Text Organisation*, TEXT 8(3), (1988), 243-281.

[8] D. Marcu, *The Theory and Practice of Discourse Parsing and Summarization*, MIT Press (2000).

[9] Mochales Palau, R., Moens, M.F. Argumentation mining, Artificial Intelligence and Law, volume 19(1), pages 1-22, 2011.

[10] P. Saint-Dizier, Processing natural language arguments with the TextCoop platform, journal of Argumentation and Computation, vol 3(1), (2012).

[11] P. Saint-Dizier, Challenges of Discourse processing: the case of technical documents, early 2014, Cambridge Scholars Publising.

[12] J. Spenader, A. Lobanova, Reliable Discourse Markers for Contrast, Eighth International Workshop on Computational Semantics, Tilburg (2007).

[13] M. Stede, *Discourse Processing*, Morgan and Claypool Publishers, (2012).

[14] M. Taboada, W. C. Mann, *Rhetorical Structure Theory: Looking back and moving ahead*, Discourse Studies, 8(3), (2006), 423-459.

[15] M. G. Villalba, P. Saint-Dizier, Some Facets of Argument Mining for Opinion Analysis , COMMA, IOS Publising, Vienna, (2012).

[16] Walton, D., Argument Mining by Applying Argumentation Schemes, Studies in Logic, Vol. 4(1), 2011.

[17] B. Webber, A. Joshi, *Discourse Structure: Past, Present and Future,* Proceedings of the ACL 2012 Workshop on Rediscovering 50 Years of Discoveries, (2012), 42-54. Jeju, Korea.

[18] G. Winterstein, *What but-sentences argue for: An argumentative analysis of but*, in Lingua 122 (2012), 1864-1885.

[19] F. Wolf, E. Gibson, *Representing Discourse Coherence: A Corpus-Based Study*, Computational Linguistics 31(2) (2005), 249-288.