# From Natural Language to Argumentation and Cognitive Systems

Theodoros Mitsikas[1], Nikolaos Spanoudakis[2], Petros Stefaneas[1], and Antonis Kakas[3]

[1] School of Applied Mathematical and Physical Sciences, National Technical University of Athens, Greece,
mitsikas@central.ntua.gr, petros@math.ntua.gr
[2] School of Production Engineering and Management, Technical University of Crete, Greece,
nikos@science.tuc.gr
[3] Department of Computer Science, University of Cyprus, Cyprus,
antonis@ucy.ac.cy

**Abstract.** This paper reports on the design and development of *Gorgias-NL*, a system based on argumentation, intended to support the development of cognitive systems. *Gorgias-NL* provides a natural language interface to the *Gorgias* argumentation system [1], aiming to capture user specifications expressed in Natural Language in the form of high-level guidelines.

The development of *Gorgias-NL* is linked with the increased interest in developing systems, such as Personal Assistants, that are cognitively compatible with the behaviour of human users, and interact directly with them in Natural Language. Driven by the natural link of argumentation to human reasoning and the computational form that it exhibits, argumentation serves as the theoretical foundation of *Gorgias-NL*. Our approach aims to produce an argumentation theory composed of arguments together with the conflicts between them, and preference information that would give strength of some arguments over others. The user's requirements input is not processed directly by some classifying or matching methods, but instead it is used to construct structures called *scenarios*, which are subsequently refined through the *SoDA* Methodology [2] to form an argumentation theory consistent with the user's preferences.

*Gorgias-NL* is based on a modular design, which is depicted in Fig. 1. Each module's function is illustrated through the following example of user input for a personal assistant managing discount coupons received by the user: *"Normally, discard coupons. If a coupon is related to my wish list, save it, unless it is expensive. If a coupon offers a large discount, save it. Discard the coupons that are out-of-date."* Firstly, the user's text input is syntactically analysed by the NLP module, which currently utilises a Stanford NLP back-end. Afterwards, the scenario generation module extracts the first level symbolic information by translating syntactic structures to symbolic predicates and atomic literals. The aforementioned example yields the following scenarios:

$< 1, 1, \{\}, discard(Coupon) >$
$< 2, 2, \{related\_to(Coupon, wish\_list)\}, save(Coupon) >$
$< 3, 2, \{related\_to(Coupon, wish\_list), expensive(Coupon)\}, neg(save(Coupon)) >$
$< 4, 3, \{large(discount), offer(Coupon, discount)\}, save(Coupon) >$
$< 5, 4, \{out\_of\_date(Coupon)\}, discard(Coupon) >$

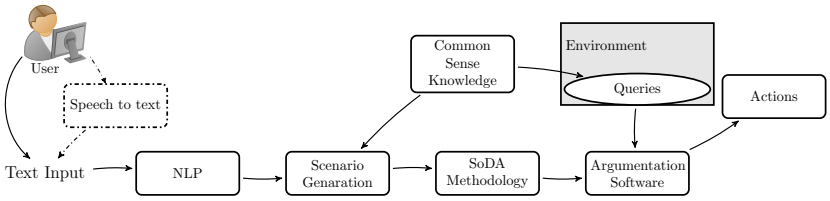


**Fig. 1.** Design of the prototype system.

The first number is the running id of the scenario whilst the second element of each scenario corresponds to the natural language sentences the scenario refers to.

The scenario generation is followed by the application of the *SoDA* Methodology, to obtain the refined scenarios which further capture the general user preferences. This is done by considering combinations of the primary scenarios which are in conflict and possibly exploiting further information about user guidelines. In the above example this would give:

$< 6, \{2,4\}, \{related\_to(C, wish\_list), out\_of\_date(C)\}, discard(C) >$
$< 7, \{3,4\}, \{large(discount), offer(C, discount), out\_of\_date(C)\}, discard(C) >$
$< 8, \{2,3\}, \{expensive(C), related\_to(C, wish\_list), large(discount),$
$offer(C, discount)\}, \{save(C), discard(C)\} >$
$< 9, \{2,3,4\}, \{expensive(C), related\_to(C, wish\_list)\}, large(discount)\},$
$offer(C, discount), out\_of\_date(C)\}, discard(C) >$

The argumentation theory that captures the user requirements from its Natural Language input then results automatically from the application of an algorithm that constructs a *context graph* for the scenarios and their refinements and then transforms this into a set of *threads*. The latter considers paths from the root to a leaf for each pair of options and creates a corresponding thread by defining all the possible contexts - combinations of conditions - from the most general to the more specific, under which a decision between a pair of options is reached. This is illustrated in Fig. 2. In this ex-

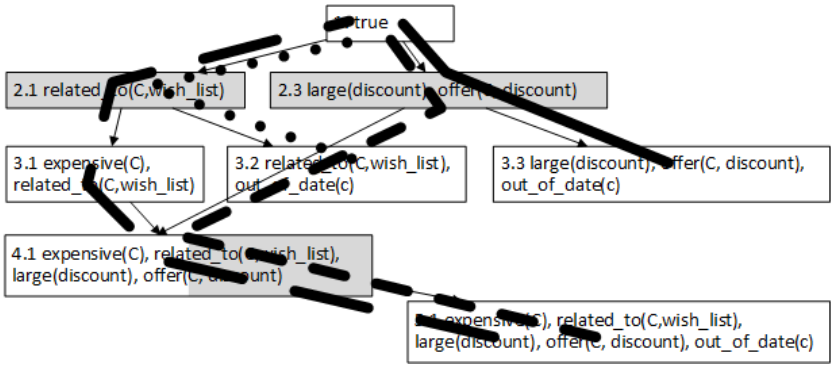


**Fig. 2.** The threads depicted graphically in the context graph.

ample, there are four threads created, where the valid option at each node is denoted by the colour of its background, white for *discard(C)* and grey for *save(C)*. For instance, the thread shown with small dashed lines, starts from the node "true". The new contextual conditions $\{large(discount), offer(C, discount)\}$ support that the option $save(C)$ is **preferred over** option $discard(C)$. More specific contextual information $\{expensive(C), related\_to(C, wish\_list)\}$ (note that node 3.2 is not included in the thread) supports that both options are valid, and the thread *forks* in two different threads. Lastly, the new contextual condition $\{out\_of\_date(C)\}$ on the forked thread supports the priority of $discard(C)$ over $save(C)$.

Future challenges include open NLP problems, while the main challenge is the incorporation, organization and utilization of Common-sense Knowledge about the concepts that are used in the natural language description of the user's requirements.

## References


1. Kakas, A., Moraitis, P.: Argumentation based decision making for autonomous agents. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems. pp. 883–890. AAMAS '03, ACM, New York, NY, USA (2003)
2. Spanoudakis, N.I., Kakas, A.C., Moraitis, P.: Applications of Argumentation: The SoDA Methodology. In: ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands. pp. 1722–1723 (2016)