Domain-Adaptive Information Extraction

Günter Neumann and **Thierry Declerck** [1]

Abstract. We present in this paper the methodology developed within the PARADIME (Parameterizable Domain-Adaptive Information and Message Extraction) project for designing an Information Extraction (IE) system easily adaptable to new domains of application. For this we went for a strict separation of the (shallow) linguistic processing modules on the one hand and the domain-modeling modules on the other hand, thus looking for the maximal degree of reusability of common linguistic resources shared by all domains of application. The tools used for the domain-modeling allow a declarative description of the domain under consideration and a simple (abstract) mapping to the output of the Natural Language (NL) analysis, thus requiring only few and very general linguistic knowledge for the adaptation of the IE-system to new applications. We describe a real scale experiment on a fast adaptation cycle of the system to a new domain – the soccer domain – and present the first results obtained.

1 Introduction

In order to overcome the problem of finding or extracting relevant information out of the enormous amount of text data electronically available, various technologies for information management systems have been explored within the Natural Language Processing (NLP) community. One line of such research is the investigation and development of *information extraction* systems.

Information extraction (IE) is the task of identifying, collecting and normalizing information from NL text. The information of interest is typically pre-specified in form of uninstantiated frame-like structures also called *templates*. The templates are domain and task specific. The major task of an IE-system is then the identification of the relevant parts of the text which are used to fill a template's slot. In order to achieve the necessary degree of robustness and efficiency, domain knowledge and shallow algorithms are used which have been tuned – usually by hand – for the actual domain and task.

IE technology has already a high degree of application potential (e.g., intelligent information retrieval, linguistically based data mining, automatic term extraction for data bases, fine-grained text classification) and has shown important industrial application impact. However, the major disadvantage of the current IE technology is that the application specific knowledge has to be directly integrated with the domain-independent linguistic knowledge sources in order to achieve the necessary degree of robustness and efficiency. This means that the system customization is based on a very low-level direct linkage of linguistic and domain specific knowledge. Since defining and implementing this low-level linkage is a time-consuming task which has to be performed manually by experts, the necessary high amount of system customization leads to a low degree of flexibility and adaptability of a system to new applications and domains.

In order to cope with this shortcoming, we started to investigate methods and technologies which support the development of IE systems which can rapidly be configured for new domains and tasks, i.e., IE systems which support a fast application development cycle. The core idea is to model the linkage of domain-independent and domain knowledge on a much higher and abstract level than it is usually done in current IE systems *without* critical compromises wrt. robustness and efficiency. More precisely, the new IE model we propose is based on:

- a domain-independent shallow text processor, which maps an NL text into a set of underspecified (partial) functional descriptions;
- modeling of domain and template definitions using type hierarchies without reference to linguistic knowledge;
- modeling of abstract linguistic types on the basis of a generalization of subcategorization frames and phrasal information;
- declarative linkage of linguistic and domain knowledge through multiple inheritance (the resulting types are called *linking types*);
- the construction of new applications basically through specialization of the domain knowledge hierarchy and specification of a domain lexicon.

The main advantages towards a faster development cycle are then:

- a high degree of modularity, re-usability and knowledge-sharing;
- template definition and specification of linking types can be separated;
- no deep internal knowledge of the system (especially about the relationship between NL and its domain specific meaning representation) is necessary for building a new application.

A prototype of the model described here – called SMES– has been fully implemented and uses broad coverage linguistic knowledge sources. Although the SMES technology has only been used up to now for handling German the underlying strategies are also applicable to other languages as well.[2]

2 Overview of the IE-Model of SMES

As mentioned above, the new IE model proposed in the PARADIME project is based on a systematic separation operated between the NLP components, dealing with the domain-independent general linguistic knowledge, and the domain modeling components, handling the domain specific knowledge. A declarative linkage of linguistic and domain knowledge is then provided by an abstract mapping. An instantiation of the overall architecture, showing the flow of information between the main components (the boxes in the graphic) is given in

[1] DFKI GmbH, German Research Center for Artificial Intelligence, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, email: {neumann,declerck}@dfki.de

[2] First versions for English and Japanese have already been realized.

figure 3. Looking at the system in a more detailed way, one can distinguish four main modules, described in the following sections.

2.1 The Shallow Text Processor (STP)

The general *linguistic analysis* is performed by a set of integrated tools supporting partial and shallow text processing. This set of tools comprise:

1. a generic lexicon (with more than 150,000 morpho-syntactically marked stems), including a database containing valence information for 12,000 verbs, and large specialized lexicons (gazetteers).
2. a tokenizer, a morphological analyzer (including on-line compound analysis) and a POS filter for the lexical processing.
3. a fragment recognizer for Named Entities and generic phrases (NP, PP, Verbgroup), see figure 1.
4. On the top of the fragment recognizer, a dependency based parser computes at the clause level a flat (partial) analysis of the text, enriched with information about grammatical functions (figure 1), also called *underspecified (partial) functional descriptions* UFD[3].
5. Completing this processing chain, an algorithm for the resolution of references – dealing both with anaphora and ellipsis resolution – is actually being implemented.

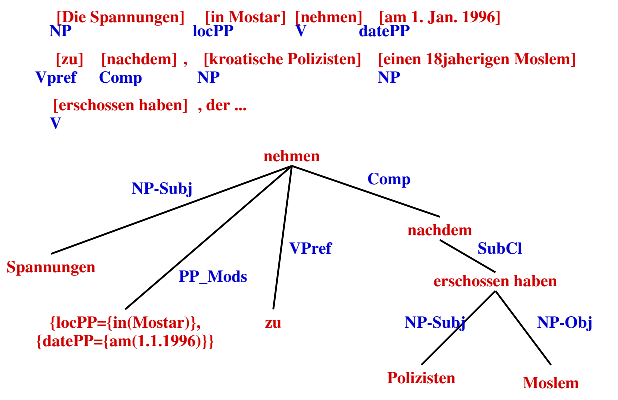

Figure 1. The output of the fragment recognizer (above) and of the (flat) dependency parser, represented as a tree

The STP component has a high degree of coverage on unseen data[4] and has a very good run-time performance (less than 1 second/page on standard PC-hardware).

The (shallow) linguistic analysis is kept generic, the claim being that the NLP tools should be as few as possible concerned by the adaptation of the IE machinery to a new application[5]. Keeping this integrated set of tools independent of the other modules of the system allows also to separately continue to improve the linguistic processing, which is quite important since complex IE is only possible on the basis of high quality and broad linguistic analysis.

2.2 The Domain Modeling Component

The *domain model* is realized by hierarchically organized typed feature structures (templates) representing the information to be extracted from text (see the right box in the left frame in figure 2). The formalism used for defining those templates is the Type Description Language (TDL)[6], which supports all the major operations on (typed) feature structures. Using the structure-sharing and the multiple inheritance facilities of TDL, we associate with the domain model strictly speaking some conceptual hierarchies abstracting over the results of the NL analysis (in our example in figure 2, a combination of hierarchies of phrasal elements and of functional descriptions). This allows the formulation of an abstract linkage between domain-independent and domain knowledge, at every level of the distinct hierarchies described. Those information-sharing links – also called *linking types*[7] – describe the underlying structure of the templates to be selected and filled with specific results of the NL processing.

An advantage of using a well-known and well-defined high-level linguistic formalism for modeling the domain knowledge is the fact that it allows us to declaratively formulate at a very abstract level semantic and pragmactic *constraints* on the output of the shallow syntactic processing.[8]

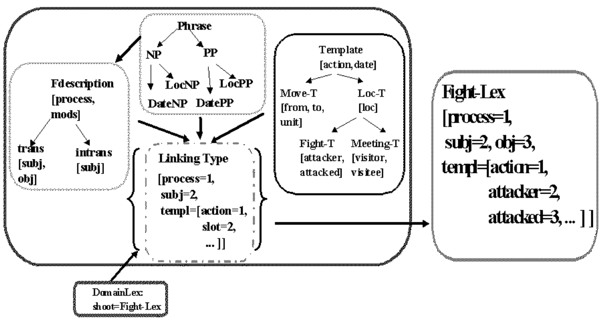

Figure 2. The domain model: in the left frame one can see the combination of conceptual hierarchies (domain and linguistic knowledge) used for defining a set of linking types. Mapping an entry of the domain lexicon into this set generates the uninstantiated template to be filled

2.3 The Domain Lexicon

It is quite typical for IE-systems to define and use lexical anchors for supporting the detection of the relevant information in text. Those

[3] An UFD is underspecified in the sense that attachments are not yet resolved (avoiding in that way the cost of massive overgeneration, due to the lack of very sophisticated lexical information at this level); rather, a kind of *upper bounds* over different possibilities for attachment and scoping of modifiers are expressed, to be further specified — whenever possible — only in the light of the domain-specific knowledge model at a later stage.

[4] An evaluation of a former application of the system (based on a subset of 35.000 tokens from the German weekly newspaper "Wirtschaftswoche") yield: From the 93,89% of the tokens which were identified by the morphological component as valid word forms, 94,37% got a unique POS-assignment with an accuracy of 97,9%. The named-entity recognition yielded a F-measure of 90.1%. For the phrase recognizer we obtained a F-measure of 87.8% and for the dependency based parser a F-measure of 87.14% on a subset of 6306 tokens (400 sentences).

[5] In fact the adaptation of the IE-system to a new domain can also be understood as providing a new *interpretation* to some of the data provided by the generic linguistic analysis, as we will see later in more details.

[6] See [3]

[7] This naming allows also to distinguish between the templates used for the domain modeling strictly speaking and the templates described by the abstract linkage, which are in fact including the former kind of templates.

[8] And this gives us in the longer term a practicable way of integrating shallow processing and methods for deep linguistic analysis.

anchors are normally listed in domain lexicons. In SMES we are using for the time being only verbs as anchor, and thus the current IE machinery is restricted to the level of verbal projections[9]. The domain lexicon associates domain-relevant verbal entries with specific templates of some domain (more specifically with the corresponding name of a linking type), see figure 2.

2.4 The Template Generator

Each UFD computed by STP is passed to the template generator which combines relevant parts of the knowledge represented in the domain model via the domain lexicon in order to end out with a set of filled templates. This is actually done by *merging* the information of the domain lexicon (linked types) with the UFDs and by mapping the whole structure into a still uninstantiated (partial) TDL instance. *Template filling* is now automatically done via TDL-based type expansion. In this way the contraints of the relevant linking types will direct the flow of information between the grammatical and domain relations (see figure 3). For the filling of those domain relations which are only constrained by type restrictions (e.g. domain-specific modifiers), we use domain-specific type inference rules.

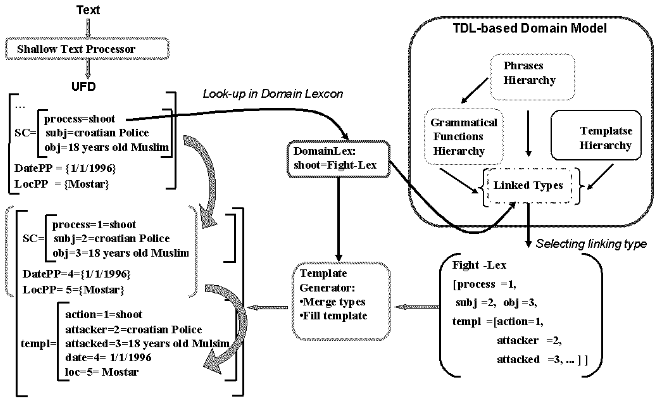

Figure 3. A sketch of the whole processain chain leading to template filling

3 Adaptation of the IE-system

Our system has already been successfully used in a number of different applications. The experiment done in PARADIME on fast development cycle, under the consideration of the maximal reusability of resources, was meant to clarify the following emerging issues:

1. What are the steps involved in such an adaptation?
2. Which are the modules involved by such an adaptation?
3. How fast can such an adaptation be?

It appears that the answering to those questions has to take into account the kind of IE subtask under consideration. Typically, IE is subdivided in distinct – but at some level interacting – subtasks[10]:

- Named Entity task (NE): Mark into the text each string that represents, a person, organization, or location name, or a date or time, or a currency or percentage figure etc.;
- Template Element task (TE): Extract information related to organization, person, and other entities, drawing evidence from everywhere in text (TE consists in generic objects and slots for a given scenario, but is unconcerned with relevance for this scenario);
- Template Relation task (TR): Extract relational information on employee_of, location_of relations etc. (TR expresses domain-independent relationships between entities identified by TE);
- Scenario Template task (ST): Extract prespecified event information and relate the event information to particular organization, person, or artifact entities (ST identifies domain and task specific entities and relations);
- Coreference task (CO): Capture information on corefering expressions, i.e. all mentions of a given entity, including those marked in NE and TE.[11]

3.1 Steps for the Adaptation

For adapting the IE-system to a new domain (in our experiment, the soccer domain), at least three processing steps are necessary:

1. Data collection, corpus and domain analysis, identification of typical terms, relations and events, and description of the templates to be filled for the application. This task is a constant one for every adaptation to new domains (can be tackled by the user or by the developer, or a combination of both). The efficiency and accuracy of this task depends on the expertise of the persons and on the quality of the tools involved. We are for example starting to investigate the deployment of corpus linguistics methods (combined with our rule-based NL processing tools) in order to speed up and improve the detection of domain relevant words and structures and so automatically supporting the building of the domain-lexicon and the domain-specific ontology.
2. Integration of the templates into a conceptual hierarchy (ontology) in order to describe the domain model and (partially) merge this conceptual structure into existing ontologies. This is the basis of the definition the linking types for template filling. The linking types will have to define specific *interpretations* of the data delivered by the generic NLP tools and also describe semantic and pragmatic *contraints* on this output in order to ensure the accuracy of the template selection and filling. This taks is eased in our case through the use of a typed feature formalism.
3. Selective adaptation of the modules of the NLP component of the IE, if necessary, and description of the domain lexicon (containing at least the typical event words). Ideally no module of the NLP component should be affected, and so the improvement work made necessary by a new application will tend to make those tools still more generic.

3.2 The Real Scale Experiment

The concrete adaptation of the IE-system to the soccer domain was then following the three steps mentioned in section 3.1, according to the sub-tasks defined in MUC-7.

In step 1) we have been collecting 323 texts about the Soccer World Championship 1998 from the Frankfurter Rundschau (German newspaper available on-line) out of which the game reports (74

[9] This actual restriction is motivated only by time constraints given to the project.
[10] The following (standard) listing of IE-subtasks is the one given by the MUC-7 conference [4].
[11] As a reminder: this task, and also the resolution of ellipsis, is currently being implemented in our project, so we will not say anything substantial about this here.

texts) have been selected for detailed corpus analysis[12]. The analysis allowed to detect domain specific terms, relations and events:

- Terms as descriptors for the NE task (more fine-grained as in MUC-7) – TEAM: *Titelverteidiger* Brasilien; PLAYER: *Superstar* Ronaldo, von *Bewacher* Calderwood noch von *Abwehrchef* Hendry; REFEREE: vom spanischen *Schiedsrichter* Garcia Aranda; TRAINER: Schottlands *Trainer Brown*; Location: *im Stade* de France; ATTENDANCE: Vor 80000 *Zuschauer*;
- Terms for NE Task – TIME: *in der 73. Minute*, von Roberto Carlos *(16.)*, scheiterte Rivaldo *(49./52.)*; DATE: *am Mittwoch*; SCORE/RESULT: Brasilien besiegt Schottland *2:1*, einen *2:1 (1:1)-Sieg*, der zwischenzeitliche *Ausgleich*;
- Relations for TR Task – OPPONENTS: *Brasilien* besiegt *Schottland*, feierte *der Top-Favorit* ... einen glücklichen 2:1 (1:1)-Sieg ber den *respektlosen Aussenseiter Schottland*; PLAYER_OF: hatte *Cesar Sampaio den vierfachen Weltmeister* ... in Führung gebracht, *Collins* gelang ... der zwischenzeitliche Ausgleich für *die Schotten*; TRAINER_OF: *Schottlands* Trainer *Brown*;
- Events for ST task – GOAL: in der 4. Minute *in Führung gebracht*, das schnellste Tor ... *markiert*, Cesar Sampaio *köpfte* zum 1:0 *ein*; FOUL: als er den durchlaufenden Gallacher im Strafraum allzu energisch am Trikot *zog*; SUBSTITUTION: und musste in der 59. Minute für Crespo *Platz machen*.

In step 2) the templates for the soccer domain has been defined on the base of the corpus analysis and the design of the NE, TR, and ST tasks. The values of the individual attributes of the feature structures can be constrained to being an atom, a list, a set or another template. The relations between the attributes are implicitly encoded in the hierarchy of domain-specific objects and events (an appropriate typing of those structures can make the relations explicit). An example of a soccer *entity* template is giving in figure 4, showing also how this specific entity template is being embedded in an *event* template, where *information-sharing* is provided for the value of identical attributes at distinct levels of embedding. The level of embedding itself is depending on the domain and the detail of the corpus analysis. In our case the top level is the one of a game of the championship, being identified by the date and the opponents involved (see figure 4).

For the third step defined in section 3.1 we defined the *linking types* on the basis of the classification of the domain-specific verbs detected by the corpus analysis, taking into account the various verb frames, the polarity and the realization of certain modifiers within the sentential clauses under consideration. The top level linking type is called *soccer-lex* and associates at the abstract level a domain-specific verb with the general template *wm98-template* (first example in figure 5). On the basis of further properties of the verb and the associated linguistic material within the clause boundaries, a subtyping of the main linking type is provided. So for a verb referring to a “game result”, a linking type *soccer-result-lex* has been introduced corresponding to the classification of such verbs in the domain-lexicon. So for example: “entry=besieg, cat=v, dom=soccer, type=goal-subj-obj”) where also the subcategorization information is playing a role, defining thus a further subtyping in the linking type hierarchy: *soccer-result-subj-obj-lex*. For the final template filling (can also be considered as a template instantiation), the subject and the object detected by the NLP tools will be associated with the respective opponents slots of the *wm98-template* (see figure 4). Not only verb arguments are taken into consideration for filling the domain

[12] Actually only 62 texts have been considered for corpus anlysis and the remaining 12 will serve as the test corpus.

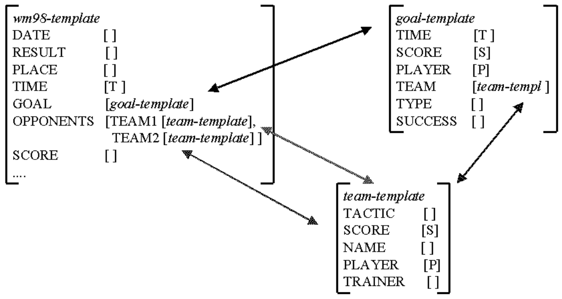

Figure 4. An *entity* template for the soccer domain: the TEAM-template and its embedding in various level of template definition (*event* and *scenario* templates), where the information-sharing is represented by variables

templates, also adjuncts are playing a role for describing relations between entities, and for example lexically restricted PP modifications can be selected for detecting a template relevant entity.

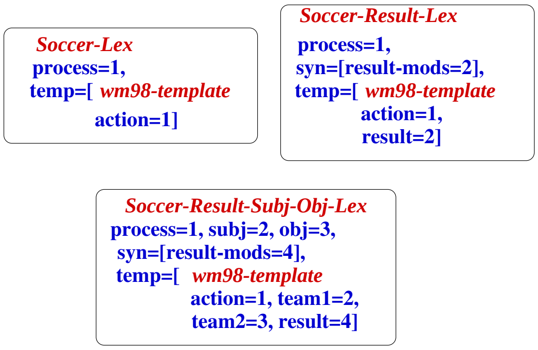

Figure 5. An example of hierarchy of linking types

This process of associating certain linguistic structures to selected attributes of templates, is what we mean when we speak of the domain specific *interpretation* of the generic data delivered by the NL processing. So in the case of the soccer domain, the linking types will interprete certain linguistic data (subject, object, etc.) as being a team, or a trainer, or a player or an attendance etc. in dependence of the template selected by the look up in the domain lexicon and of the consideration of certain constraints formulated in the conceptual hierarchy.

3.3 System Modules concerned by the Adaptation

For the NE task some modification has been done in the components of the NLP tools, consisting in the description of new patterns for the detection of special Named Entities. But no modification of the components dealing with the recognition of generic phrase was necessary. Since the domain lexicon has an interface to the generic lexicon, the latter had to be extended for some verbs not covered yet. This was a mimimal adjustement. The rest of the adaptation work has been indeed exclusively done in the context of the domain modeling.

3.4 Required Time for the Adaptation

Since our expectation that the adaptation of the IE-system can basically be done through the sole design of an appropriate conceptual hierarchy, we could concentrate our effort on this design and the development cycle for a first prototype for a new application has been substantially speeded up: twice as fast as similar experiences with our former system, where the linkage of domain-knowledge and linguistic processing was done at a low level. With the new IE model, we can say that a running prototype for a new application can be designed with the effort of one man/month, once the corpus analysis has been provided. The quality of this prototype, discussed in the next section about the evaluation, can still be improved by further development cycles done an both the NLP components[13] and some refinements of the domain modeling.

4 Evaluation

For the (blind) testing of our adaptation work done sofar, we processed the 12 texts of the test corpus. The metrics we adopted are the one typically used for IE, i.e. precision (P), recall (R) and F-measure (combined P and R). The results are presented in table 1, where percentages are given for each IE-subtasks (none for the CO subtask in our case), which are described in section 3 above:

Table 1. In the following table, the results of the real scale experiment are given.

Task	Recall	Precision	R&P
NE	85.71	90	87.80
CO	-	-	-
TE	59.09	81.25	68.35
TR	45	64	52.84
ST	42.59	62.16	50.54

Those results have to be compared with the ones proposed by some of the participants of the MUC-7 conference, in fact the comparison can be done only with the system described by the University of Sheffield, since they were the only group concerned with all the subtasks defined for the MUC-7 conference[14]:

Table 2. In the following table, we give the results of the system of the MUC-7 participant University of Sheffield

Task	Recall	Precision	R&P
NE	83	89	85.83
CO	56.1	68.8	61.8
TE	75	80	77.17
TR	41	82	54.70
ST	47	42	44.04

The systems have quite similar results on the NE task. Our sytem is being better on the detection of scenario templates, but it might be that the restriction of our system to verbal constructs leads to this result (the evaluation being sensitive to this restriction). Since our system didn't cope with CO task (and ellipsis resolution) our TE results are not as good as they could be. Improvement for the TR task can be expected if we take into account all possessive constructions in the text, which we didn't consider in out *linking tyoes*.

But in general, and despite of the language distinction, we can see that both systems show similar results, which is quite promising for our approach, considering that the development cycle we had was quite short, and improvements are still to be expected in dependency of furhter developments of the linguistic processing.

5 Related Work

We are not aware of any other German IE system which has a comparable coverage and performance on the linguistic side as well as a similar concise modeling of IE applications via domain modeling using advanced typed feature formalism. The generic architecture of the shallow text processor is similar to most well-known IE systems (see MUC-7), modulo language aspects. However the modeling of linguistic and domain knowledge and their interaction at an abstract level is quite novel in the case of IE. Here the domain modeling is seen from an engineering point of view, following a bottow-up knowledge specification, in contrast to approaches of ontology which try to establish generic domain-independent constraints of lexical semantics following a top-down approach (for example the Upper Model [1]).

6 Conclusion

We have shown that the new IE model we investigated and developed is indeed suitable for a faster adaptation of the IE machinery to a new domain of application (compared both with the time necessary for a development cycle with our former system and with the results obtained by participants of the MUC-7 conference). Our future work will be concerned with the use of corpus linguistic methods for speeding up the domain modeling task, giving automatic support for the building of domain lexicons. We will also investigate in an European Consortium the multinlingual extension of our IE system, where we expect the domain model to be stable with repect to the distinct languages, only the linking types being concerned by the variety of languages.

ACKNOWLEDGEMENTS

The research underlying this paper was supported by grants from the German Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMB+F) to the DFKI project PARADIME (FKZ ITW 9704). We are very grateful to Giampaolo Mazzini (Celi, Torino) who contributed a lot to this research as he was a guest researcher at DFKI in the year 1998.

REFERENCES

[1] J. A. Bateman, `Ontology construction and natural language', in *Proceedings of the International Workshop on Formal Ontology*, (1993).

[2] K. et al. Humphreys, `University of sheffield: Descritpion of the lasie-ii system as used for muc-7', in *Proceedings of the Seventh Message Understanding Conference (MUC-7)y*, ed., SAIC, http://www.muc.saic.com/, (1998). SAIC Information Extraction.

[3] Hans-Ulrich Krieger and Ulrich Schäfer, `$\mathcal{TDL}$—a type description language for constraint-based grammars', in *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*, pp. 893–899, (1994).

[4] SAIC, ed. *Seventh Message Understanding Conference (MUC-7)*, http://www.muc.saic.com/, 1998. SAIC Information Extraction.

[13] In our concrete case, adding the processing step concerned with reference and ellipsis resolution

[14] See [2]