Semantically Driven Approach for Scenario Recognition in the IE System FRET

Svetla Boytcheva[1], Milena Yankova[2] and Albena Strupchanska[2]

[1]Department of Information Technologies,
Faculty of Mathematics & Informatics
Sofia University "St. Kl. Ohridski",
5 James Bauchier Str.,
1164 Sofia, Bulgaria
svetla@fmi.uni-sofia.bg

[2]Linguistic Modelling Department,
Central Laboratory for Parallel Processing,
Bulgarian Academy of Sciences,
25A Acad. G. Bonchev Str.,
1113 Sofia, Bulgaria
{myankova, albena}@lml.bas.bg

Abstract

This paper reports a research effort in scenario recognition task in information extraction. The presented approach uses partial semantic analysis based on logical form representation of the templates and the processed text. It is implemented in the system FRET[1] (Football Reports Extraction Templates), which processes specific temporally structured texts. A logical inference mechanism is used for filling template forms. Only scenario-relevant relations between events are linked in the inference chains. The knowledge base plays an important role in this process. Some aspects of negation and modalities that occur in the texts are also taken into account.

1. Introduction

The most important and difficult task in Information Extraction (IE) is scenario recognition. There are many systems that address this problem applying different approaches. Roughly, these approaches can be characterised as shallow or deep depending on the processing level at each of the stages. The variety can be found in the syntactic analysis, which varies from phrasal chunking to parsing and produces regularised forms. They can be anything from partially filled templates to full logical forms. On the one hand there are systems, which apply domain-specific lexically triggered patterns and on the other hand there are systems, which employ complete parsers for context-free or even more expressive formalisms to apply general grammars of natural language.

The discourse or multi-sentence level processing that follows the syntactic analysis can be also more or less deep. It depends on the usage of declaratively represented world and domain knowledge to help resolving ambiguities of attachments, word sense, quantifiers scope, and coreferences or to support inference-driven templates filling [7]. One of the last presented systems that have domain specific knowledge as semantic net is LaSIE [6, 9], and it attempts fragmentary parsing only and falls somewhere in between the deep and the shallow approaches.

Most of the systems presented in the latest MUC (Message Understanding Conferences) avoid usage of deep processing mainly because it is hard to accomplish many of the required natural language understanding tasks. Some authors [8] believe that only grammatical relations relevant to the template should be considered. The great effort needed for building up the domain knowledge supporting the language understanding is another reason for using shallow processing.

FASTUS [1, 2] is a good example for the shallow approach implementation. It provides phrase parser and recognises domain patterns for building raw templates, which are normalised by the postprocessor. However, even when FASTUS showed second best results on MUC-5, the authors reported on difficulties in defining all possible patterns for a given event.

[1] This work is partially supported by the European Commission via contract ICA1-2000-70016 "BIS-21 Centre of Excellence"

That's why we went backward to the systems for reinventing the deeper techniques for scenario matching, but it seems to be too time and effort consuming to provide complete parsing, disambiguation etc., so we decided to stay somewhere in the middle.

Our system focuses on templates filling because we consider it as an important and still open question. There are many systems [14] for automatic or semi-automatic generation of templates based on machine learning approach. However, we assume that all templates are given beforehand and the system tries only to recognise scenario and to fill the corresponding templates correctly.

In this paper we present semantically driven approach for scenario pattern matching in the IE system FRET [15]. Our approach is to provide deep understanding only in "certain scenario-relevant points" by elaborating the inference mechanisms.

The paper is organised as follows: Section 2 presents a short overview of FRET architecture as a whole. Section 3 discuses our improvement in the translation to logical form, especially the coreference resolution and the recognition of negation and modalities, which appear in the chosen domain. Section 4 describes the inference mechanism integrated in FRET. Evaluation results are in Section 5. Section 6 contains the conclusion and sketches some directions for further work.

2. FRET Architecture

Recognising scenarios is a hard IE task in a frequently changeable domain as the football one. Both the domain terminology and statements in football reports are fast changing. There is no certainty in the truth of the stated facts as they could be negated later. The specific usage of words, which are treated as terms in the domain, is embarrassing even for human beings. Also it is worthwhile to mention that football reports have paragraph structure with tickers for each minute. So the reports provide rich temporal information that simplifies the choice of text parts to be processed in search of scenario. All these factors have to be taken into account during the development of the Knowledge Base (KB) and the system architecture.

The system consists of three main modules (Fig.1): text preprocessor, logical form translator and templates filler. The text preprocessor performs lexical analysis, Name Entity (NE) recognition and part-of-speech tagging of football reports. GATE system [4] is integrated in FRET and its modules perform these tasks.

The coreference resolution task is performed by the logical form translator. Its algorithm is developed taking into account the domain specificity. However, it is very difficult to cope with the usage of nicknames of football players and the variety of foreign names with their transcriptions. These problems reflect on the performance of the NE recognition and respectively on the coreference resolution tasks. Another problem that we do not solve is the usage of metaphors.

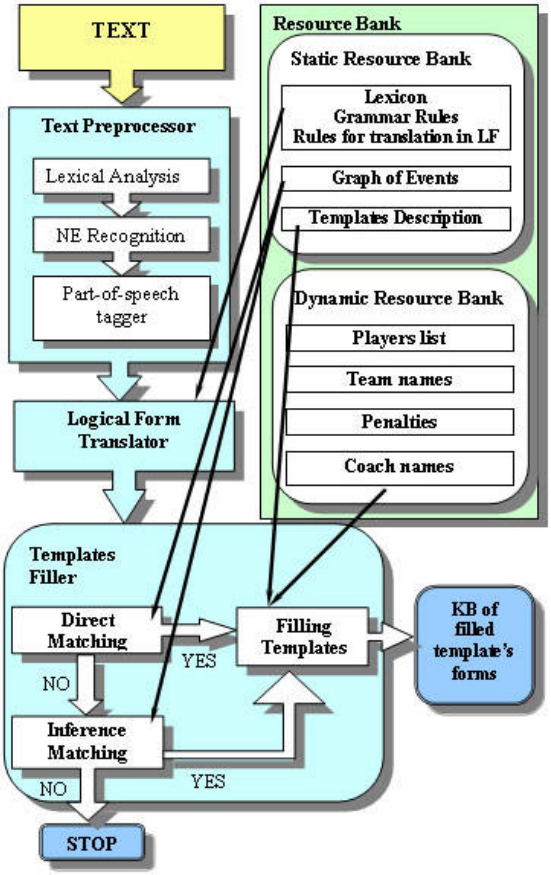

Figure 1. FRET Architecture

The KB of FRET contains two main parts: static and dynamic resource banks. The static resource bank includes lexicon, grammar rules, rules for translation in Logical Forms (LFs), templates description and description of the domain events with their relations. All uninstantiated events LFs and relations between them are presented respectively as a graph nodes and arcs. More details about the types of events and relations are given in Section 4. Some information concerning the team/coach names, players' list, playing roles, penalties, etc. is not constant so it could not be added to the static part of the KB. It is included into the dynamic resource bank and is automatically collected for each football report during the text processing. Both the logical form translator and the templates filler use the KB.

3. Logical form translation

Usually in NL texts, fragments of partial information about an event are spread over several sentences. The descriptions need to be combined before the scenario recognition. That's why FRET associates the time of the event to each produced LF. Every LF is decomposed into its disjuncts and each of them is marked with the associated time.

A specially developed partial syntactic parser implemented in Sicstus Prolog is used in FRET [15] for logical form translation. All words in LF are represented as predicates, where the predicate symbol is the corresponding base form of the word and has one argument. Specially denoted predicates with a symbol "theta" and an arity 3 are used for representation of thematic roles (see Example 1). The arguments describe the thematic role and its relations with the corresponding predicates. In the case of proper name occurrence, the argument is substituted by the string of the name.

Example 1:
Sentence:
17 mins: Beckham fires the ball into Veron.
Logical form:

```
time(17) & fire(A) &
theta(A, agnt, 'Beckham') &
ball(D) & theta(A, obj, D) &
theta(A, into, 'Veron').
```

Some aspects of coreference resolution [5] are also solved on this stage. Two types of coreference are important for the discussed system. The first one is pronominal, which is the most common type of coreference in the chosen domain.

Pronominal coreference resolution is restricted according to the domain only in detection of the proper antecedent for the following pronouns:

- personal: *he, him*
- possessive: *his*

These anaphoras are solved by binding the pronoun to the nearest left position agent in the extended paragraph, which includes the last sentence form the previous minute, all sentences from the current minute and the first sentence form the next minute.

- relative: *who*

This case is solved by binding the pronoun to the nearest left position noun in the current sentence.

The second coreference type is proper names coreference, which is based on the players' lists from the dynamic resource bank.

Other problem that has to be solved during the parsing process is the identification of negations. As described in [3] and taking into account the specific domain texts, we distinguish explicit and implicit negations (they appear in our test corpus as shown in Table1).

sentences	**count**
total	*4188*
with explicit negation	*603*
with implicit negation	*1581*
with modalities	*401*

Table 1. Occurrences of negation and modalities in the corpus

We consider two types of explicit usage of negation:

- Short sentence containing only "No". In this case we mark the LF of previous sentence with marker for negation "NEG";

- Complete sentence, containing "Not/Non/No". In this case the scope of the negation is the succeeding part of the current sentence and thus we mark only its LF with marker for negation "NEG".

In implicit usage of negation (words as *"but", "however", "although*"…) [10], both LFs of words preceding and succeeding the negation in the sentence (in some cases previous or next sentence) are marked with markers for negation: "BAHpos" and "BAHneg" (see Example 2, BAH is an abbreviation for But, Although, However).

Example 2:
Sentence:
79 mins: Henry fires at goal, but misses from a tight angle.
Logical forms:

```
time(79) & fire(A) &
theta(A,agnt,'Henry') &
theta(A,at,B) & goal(B) &
marker('BAHpos',7).

time(79) & miss(A) &
theta(A,agnt,'Henry') &
theta(A,form,B) & angle(B) &
theta(B,char,C) & tight(C)&
marker('BAHneg',7).
```

Another problem that has to be solved is to recognise sentences with modalities (words as *"can", "should", "may", "have to",* …) (that are about 10% of the corpus, see Table 1) and to mark their LFs with marker "MOD". In this case we also mark the LF of the next sentence with the same marker, because we expect acceptance or rejection of the current modality in this sentence (see Example 3).

Example 3:
21 min: Jaap Stam will be next. Surely he has to score. GOAL!!! The ball reached the back of the net.

Logical forms:

```
time(21) & score(A) &
theta(A,agnt,'Jaap Stam') &
theta(A,char,B) & surely(B)&
marker('MOD',6).

time(21) & 'GOAL'(C) &
marker('MOD',6).

time(21) & reach(D) &
theta(D,agnt,E) &
ball(E) & theta(D,obj,F) &
net(F) & theta(F,poss,G) &
back(G).
```

All these markers are necessary for further inference since in the first step we only recognise modalities and possible negations and postpone their interpretation.

4. Inference mechanism

In FRET all templates are described as tables with two types of fields *(i) obligatory fields* and *(ii) optional fields* that have to be filled in. Both types of fields, taken as a whole, contain the key information presented in the text [12]. We state that the scenario is recognised if at least the obligatory fields are filled in, while the optional fields can be left empty.

In FRET we distinguish three types of events related to each scenario that are structured into a directed graph – preliminary stored in the static resource bank (see Fig. 2):

- *main event:* the template description as LF of obligatory and optional fields and relations between them
- *base events:* LFs of most important self-dependent events in the chosen domain.
- *sub-events*: kinds of base events that are immediately connected to the main event, i.e. there exists an arc between the nodes of the main and the sub-events.

The matching algorithm of FRET is based on the relations between events. Each of these relations is represented as an arc with associated weight in the graph. We use four types of relations, defined as follows:

- Event E2 ***invalidates*** event E1, i.e. event E2 happens after E1 and annuls it.
- Event E1 ***entails*** event E2, i.e. when E1 happens E2 always happens at the same time.
- Event E1 ***enables*** event E2, i.e. event E1 happens before the beginning of event E2 and event E1 is a precondition for E2.
- Event E2 ***is a part of*** event E1.

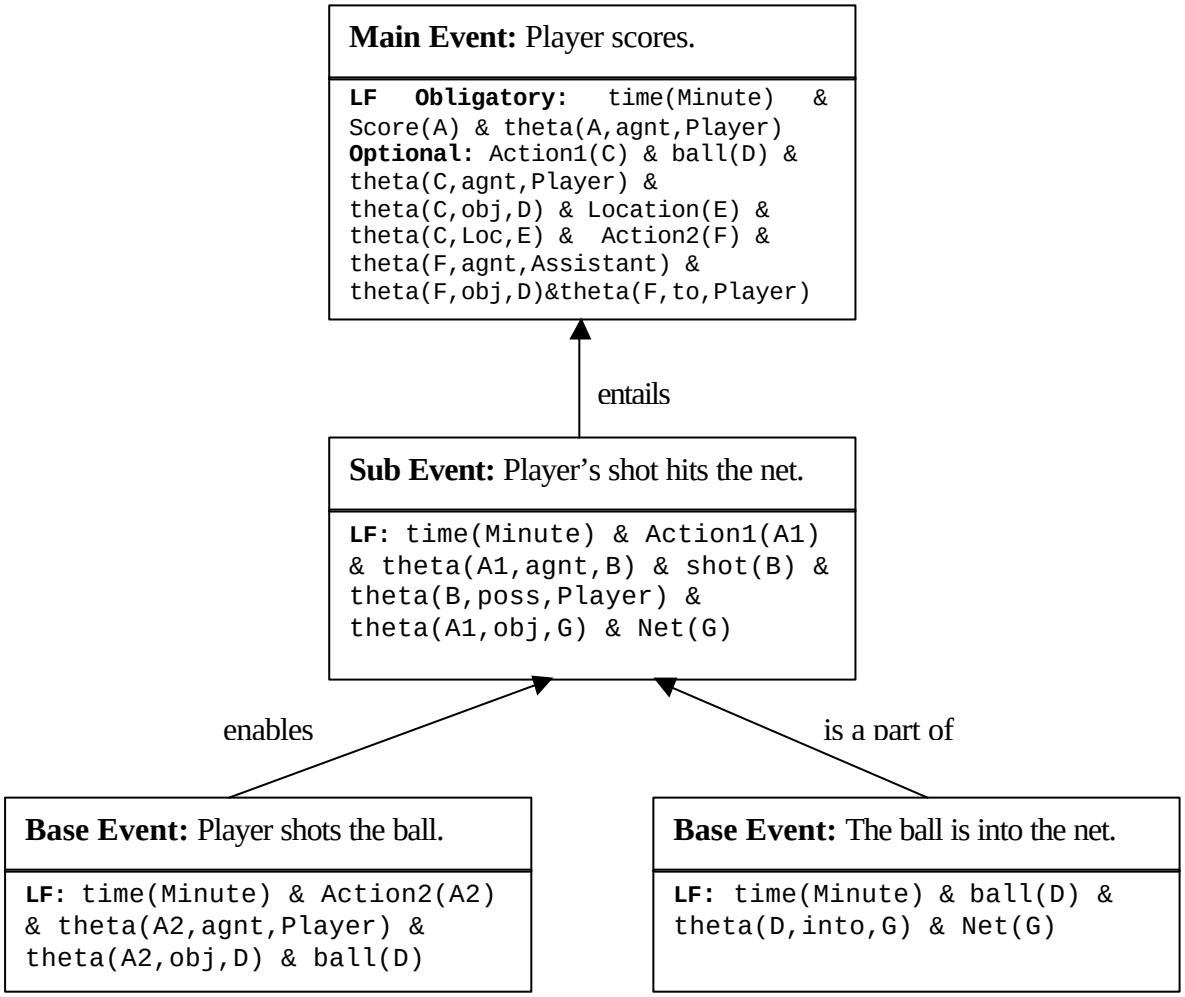

Figure 2. A part of the graph, stored in the static resource bank

The templates filling module of FRET performs two main steps:

- matching LFs;
- filling templates.

The step of matching LFs is based on the modification of the unification algorithm. We are interested in those LFs, which are produced from the so-called extended paragraph. Thus we process each paragraph separately.

Initially the module tries to match each LF from the extended paragraph to the main event. We call this step *direct matching*. The direct matching algorithm succeeds when at least one LF is matched to the main event. Then we can proceed with the filling templates algorithm. However, it starts if there are no markers for negation or modalities in the matched LF. Otherwise, the availability of such markers in the LF is an indication that there is no certainty in the truth of the statement matched to the main event. So, some additional steps are necessary for correctly processing the marked LFs and therefore recognising an event. These steps depend on the attached types of markers.

- NEG – the result from the matching algorithm is ignored, so the event is not recognised. In this case we consider that it is better not to recognise some event than to recognise it wrongly.
- BAHpos or BAHneg – we treat LFs with these markers according to the semantic interpretation of contrastive particles [13]. There are two major

interpretation types: as negation and as conjunction of independent statements. We are interested only in the contrast interpretation, that's why we have to check whether we really have negation in the marked LFs. The latter are processed as follows: all events related with *invalidate* relation to the matched event are collected into a set; the LF from the extended paragraph, which is marked with the other BAH marker is juxtaposed to a member of the collected set; if this succeeds, the previous matching is ignored.

- MOD – the event is correctly recognised only when the next LF from the extended paragraph contains explicit confirmation [11] (e.g. Yes; GOAL!!!; Bravo etc.) or the same scenario is recognised in other sentence from the current paragraph (in this case the proceeding sentence is used as additional information for filling the already matched scenario).

Direct matching algorithm succeeds when all main event's LFs variables related to obligatory fields are bound. In this step we use also synonyms lists from the KB that are necessary for the unification of variable predicate names in LFs.

If the direct matching algorithm fails FRET starts the *inference matching algorithm*. The inference matching algorithm is described below in a more formal way.

Lets make the following denotations:

EP: LFs included into the current extended paragraph;
M: the main event;
G: a corresponding graph in KB;
C: predefined coefficient.

Inference-matching algorithm

1. Construct the set $S=\{E_i : E_i$ is a sub-event of M, $i=1..n\}$;
2. $i=1$;
3. Construct the set $B=\{B_j : B_j$ is a base event from G and exists $arc(B_j,E_i,r_j)^oG\}$;
4. Apply direct matching algorithm to all possible couples (L_k, B_p) where L_k ° EP and B_p ° B. Collect successfully matched couples in the set B'.
5. Calculate $R= {}_{j=1..|B'|}r_j$ where $arc(B_j,E_i,r_j)^oG$ and $B_j{}^oB'$
6. IF R C THEN "Unify" and "Stop" ELSE i:=i+1
7. IF i n THEN goto Step 3 ELSE "Fail"

When the inference-matching algorithm succeeds, all possible predicates from the selected sub-event (E_i) are unified by corresponding variables from the predicates in the set of successfully matched base events (B'). Thus the sub-event LF is successfully matched. A similar unification is applied to the set of matched sub- and base-events ($E_i \cup B'$) and the main event (M) in order to fulfil the main event's LF (as shown at Fig. 2).

The templates filling starts only if either the direct or the inference matching algorithm succeeds.

Initially the template obligatory fields are filled by the required information. At this stage some additional information from the dynamic resource bank is used too. The completion of all obligatory fields is sufficient for correct scenario recognition. However, the optional fields, for which exist enough information, are also filled.

Example 4:

53 mins: Beckham shoots the ball across the penalty area to Alan Shearer who heads into the back of the net at the far post.

The result of text processing in Example 4 is shown at Fig 3. In this example the GOAL scenario is recognised in the paragraph marked with *"53 mins"*. The direct matching failed but the more complex inference matching algorithm succeeded in matching one of the sub-events (i.e. *"Player heads into the net"*). Thus the obligatory fields are filled with the corresponding information form the text (i.e. *player name = "Alan Shearer"*) and the

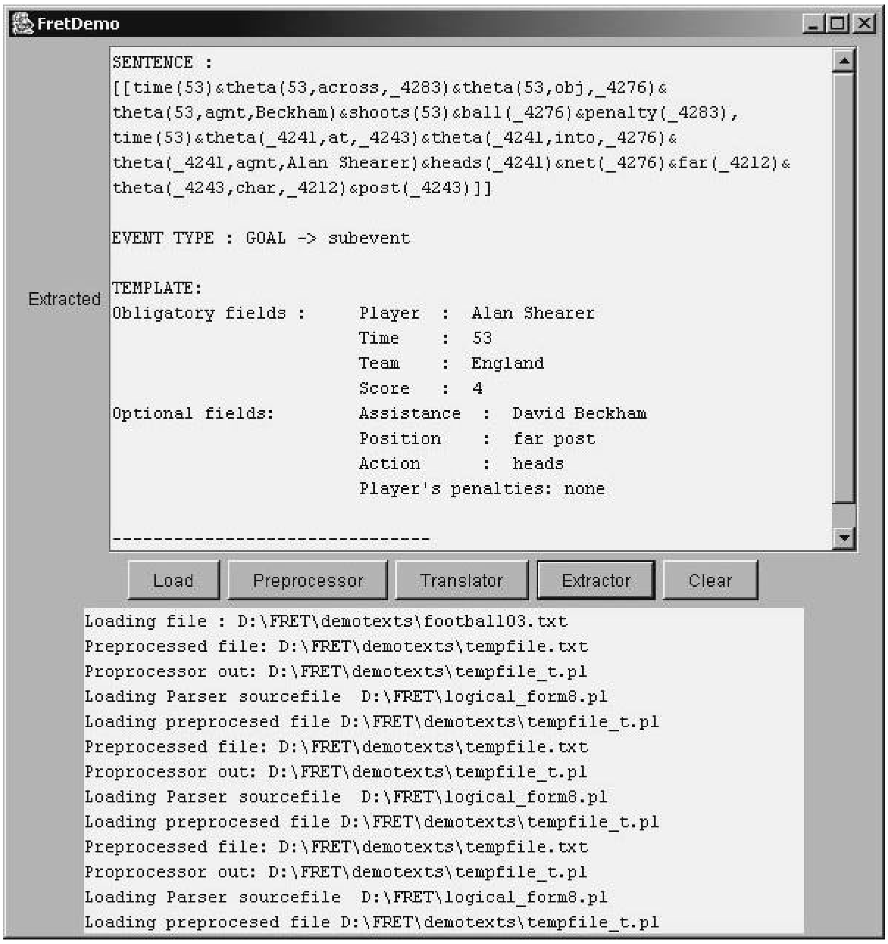

Figure 3: The result obtained after filling in a template from Example 4.

dynamic resource bank (i.e. *team* = *"England"*). Note, that in this case there is enough information for completing all optional fields but this is an optimistic case.

5. Evaluation

FRET is tested only on the scenario "goal" in the texts of 50 reports, which totally contain 148 instances of the event "goal".

The f-measure of the text processor module is smaller than 96% and depends on GATE performance. The developed algorithm for coreference resolution has f-measure < 89% (the percentage is high because we are interested only in particular cases of pronominal and proper noun coreference). The f-measure of the parser is smaller than 91%.

The domain specific treatment of negation and modalities improve the f-measure of the FRET system. So the scenario templates are filled in with precision: 86%, recall: 61 % and f-measure: 71.37 %. The direct matching works in 12% of the cases and the inference is applied in 88%.

6. Conclusion and further work

Scenario recognition is important and difficult task for information extraction. So in this paper we make an attempt to find an easy and effective way for scenario recognition that may facilitate semantic processing of large text collections.

With the usage of inference we could find either similar ways of expressing different scenarios or partial information about some event spread over several sentences. Thus we

believe that the inference is an integral part of finding facts in texts. In order to make effective inference it is necessary to represent sentences into LFs and to have suitable representation of the domain knowledge. We have to emphasise the major role of the specially tailored structure of the events and relations between them as a graph. The choice of simple relations between events makes the inference mechanism in the graph structure easier. When the simpler inference fails the more complicated one is started. However, not all the information provided in the text is needed for simple template filling. So our approach uses shallow parsing and partial semantic analysis.

The innovative aspects in FRET at this stage of development are:

- attempts for domain-specific treatment of implicit negation and modalities.
- elaborated inference mechanism that provides relatively deep NL understanding but only in "certain points". Note that the inference is simple and effective due to the consideration of only scenario relevant relations between events.

Our plans for future development of the system include making a deep investigation of the domain and completing the events graph. This will make our evaluation more precise and probably will improve the results reported above. We also plan to test the system behaviour on other domains that have a specific temporal structure.

References

[1] Appelt, D., J.Hobbs, J.Bear, D.Israel, M.Kameyama, A.Kehler, D. Mratin, K. Myers and M.Tyson (1993), "SRI International *FASTUS system: MUC-5 Test Results and Analysis*", In Proc. MUC-5, pp. 221-235.

[2] Appelt, D., J.Hobbs, J.Bear, D.Israel, M.Kameyama and M.Tyson (1995), "Description of JV-FASTUS System as Used for MUC-6", In Proc. MUC-6, pp. 237-248.

[3] Boytcheva, Sv., A. Strupchanska and G.Angelova. (July 2002), *"Processing Negation in NL Interfaces to Knowledge Bases"* In Proc. ICCS-2002, LNAI 2393, pp.137-150.

[4] Cunningaham, H., D. Mayard, K. Boncheva, V. Tablan, C. Ursu and M. Dimitrov (2002) *"The GATE User Guide*". http://gate.ac.uk/.

[5] Dimitrov, Marin (2002), *"A light-weight Approach to Coreference Resolution for Named Entities in Text"*, MSc thesis, Sofia University.

[6] Gaizauskas, R., T. Wakao, K. Humphreys, H. Cunningham, and Y. Wilks. *University of sheffield: Description of the LaSIE system as used for MUC-6*. In Proc. MUC-6. Morgan Kaufmann, 1995.

[7] Gaizauskas, R. and Y. Wilks, "*IE : Beyond Document Retrieval"*, Journal of Documentation, Vol 54, no. 1, 1998, pp. 70-105.

[8] Grishman, Ralph (1997), *"Information Extraction: Techniques and Challenges"*, International Summer School, SCIE-97.

[9] Humphreys, K., R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, Y. Wilks. *Univ. of Sheffield: Description of the LaSIE-II System as Used for MUC-7*. MUC-7. Fairfax, Virginia, 1998.

[10] Mastop, Rosja ,"*Formalizing the contrastive particle*", 2001.
http://www.geocities.com/rosjamastop/rosja_mastop.pdf

[11] Sahlqvist, H., *"Completeness and Correspondence in First and Second Order Semantics for Modal Logic"*, in Kanger, S. (ed.) Proceedings of the Third Scandanavian Logic Symposium, Amsterdam: North Holland (1975): 110-143.

[12] Wilks, Yorick (1997) *"Information Extraction as a Core Language Technology*", International Summer School, SCIE-97.

[13] Winter, Yoad and Mori Rimon (1994) *"Contrast and Implication in Natural Language"*, In Journal of Semantics 11, pp. 365-406.

[14] Yangarber, R., R.Grishman, P. Tapanainen and Silja Huttunen. *Unsupervised discovery of scenario-level patterns for information extraction*. In Proc. 6th ANLP, 2000.

[15] Yankova, M., S. Boytcheva (2003) *"Focusing on Scenario Recognition in Information Extraction"*, In Proc. EACL-2003, pp.41-48.